

国际信息工程先进技术译丛

网络编码基础与应用

Network Coding Fundamentals and Applications

（美）Muriel Médard
Alex Sprintson 等编著

郝建军 郭一珺 译





机械工业出版社
CHINA MACHINE PRESS

国际信息工程先进技术译丛

网络编码基础与应用

（美） Muriel Médard
Alex Sprintson 等编著

郝建军 郭一珺 译



机 械 工 业 出 版 社

本书由10位作者撰写而成，集各家之所长于一体，从多个角度对网络编码技术进行了深入的讲解，包括对网络编码在无线网络、对等网络、数据压缩、LTE网络、MANET、容错网络、保密与可靠通信网络、协同传输网络等场景中的研究与分析。

本书对于想要掌握该领域的基础知识并了解该技术的应用的网络专业人员来说是一个理想的选择。对于想要进行网络编码研究的研究生来说，本书也具有重要价值。本书可以作为重点在于网络编码或先进编码技术的课程的教科书或参考书。

Network Coding Fundamentals and Applications
Muriel Médard, Alex Sprintson
ISBN: 9780123809186

图书在版编目（CIP）数据

网络编码基础与应用/（美）莫德尔等编著；郝建军，郭一珺译．—北京：机械工业出版社，2013.10
（国际信息工程先进技术译丛）
ISBN 978-7-111-44506-7

Ⅰ.①网… Ⅱ.①莫…②郝…③郭… Ⅲ.①计算机网络－编码程序－程序设计 Ⅳ.①TP393

中国版本图书馆CIP数据核字（2013）第251017号

机械工业出版社（北京市百万庄大街22号 邮政编码100037）
策划编辑：李馨馨 责任编辑：李馨馨 韩 静
版式设计：霍永明 责任校对：姜 婷
封面设计：赵颖喆 责任印制：张 楠
唐山丰电印务有限公司印刷
2014年1月第1版第1次印刷
169mm×239mm · 14印张 · 308千字
0001—3000册
标准书号：ISBN 978-7-111-44506-7
定价：59.00元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

电话服务	网络服务
社服务中心：(010)88361066	教材网：http://www.cmpedu.com
销售一部：(010)68326294	机工官网：http://www.cmpbook.com
销售二部：(010)88379649	机工官博：http://weibo.com/cmp1952
读者购书热线：(010)88379203	封面无防伪标均为盗版

前　言

网络编码思想始创于千年更替之际，由 Ahlswede、蔡和杨在其开创性论文中提出后迅速吸引了世界各地的研究者和实践者。最近的研究工作阐明了网络编码的许多令人关注的问题、基本容量限制以及性能增益。对于一些重要的实际设定，研究者们设计了可达容量的网络编码，掌握了一些基本的网络编码问题的计算复杂度，并制定了有效的网络编码算法。此外，研究者们还揭示了网络编码和诸如网络、复杂理论、图论、拟阵论、编码理论以及信息论等一些其他领域之间的深刻联系。

尽管对网络编码的研究在理论方面已经取得了重大进展，但在实际应用方面才刚刚起步。虽然网络编码已经被成功地运用到了许多领域，例如最常见的内容分发系统，但对于无线网络、存储网络、容断网络等其他领域，网络编码在提升性能方面尚有许多潜力待开发。此外，网络编码技术在提高无线环境和有线环境中的安全性、鲁棒性、可管理性以及对 QoS 的支持性方面也有着巨大的潜力。

本书提供了网络编码的入门教程以及在网络和分布式计算的不同领域中的各种实际应用的综合考察。我们的目标是发现网络编码应用的丰富性并寻找激动人心的新的研究机会。全书共分 10 章，每一章均由网络编码领域公认的专家撰写，并聚焦于一个特定的应用领域。本书力求表述清晰，且使用最少的数学符号，使每位熟悉基本网络概念的读者均可以理解本书。为了便于之后的阅读，每章中作者都给出了一个全面的文献综述，该综述参考了与该领域最相关的研究工作。

第 1 章以教程的形式介绍了网络编码的基本原理，并概述了网络信息流、线性网络编码、码构造的多项式时间算法、随机网络编码、编码优势、子空间传输以及非线性网络码。

第 2 章聚焦于在无线网络中实现网络编码的基本原则，讨论了机会监听和编码的原理，详细描述了转发架构和网络编码的性能增益，还阐述了模拟网络编码的基本原理。

第 3 章着重于内容分发和多媒体流，讨论了网络编码在对等网络中的应用，探讨了在运用网络编码的对等系统中的可能的设计空间，并展示了在该设定下使用编码技术的优势。

第 4 章讨论了网络编码在商业移动平台上的实现，描述了系统的基本构成和设计选择，并讨论了与在移动电话上实现网络编码算法相关的实际问题和性能问题。

第 5 章讨论了网络编码在 LTE 网络中的应用。作者重点介绍用于消失恢复的用户协同方案，并展示了利用网络编码技术带来的显著的性能增益。

第 6 章重点关注移动 Ad Hoc 网络，并描述了一个基于网络编码协议栈的完全实现的通信系统，考察了与在嵌入式系统上实现网络编码相关的一些实际实现问题，并给出

了现场测试的结果。

第 7 章聚焦于网络编码在提升网络安全方面的应用，考虑了只有窃听能力的被动攻击以及能够同时进行窃听和干扰的主动攻击，还讨论了攻击者计算能力有限的情况。

第 8 章讨论了网络编码、源编码和数据压缩之间的关系。

第 9 章重点讨论了当对无线网络的网络规模、编码窗口大小、网络信息流数量以及应用延迟约束进行一些缩放时，网络编码的性能变化，推导并分析了有助于量化网络编码在大型网络中的性能的标度律。

第 10 章聚焦于网络编码在延迟和容错网络中的应用，讨论了在设计这类网络时面临的独特的挑战，并证明了网络编码技术有助于解决这些挑战，讨论了传输和缓存管理、恢复方案以及在能量效率和提升延迟-传输权衡方面的编码效益。

目标读者

本书是为对网络编码的一般领域以及在通信网络的不同领域上的应用感兴趣的研究者和实践者们准备的。我们假定本书的读者只具备一般的网络技术背景，且之前没有接触过网络编码技术或网络编码应用。本书对于想要掌握该领域的基础知识并了解该技术的应用的网络专业人员来说是一个理想的选择。对于想要进行网络编码研究的研究生来说，本书也具有重要价值。本书可以作为重点在于网络编码或先进编码技术的课程的教科书或参考书。

Muriel Medard

美国马萨诸塞州，剑桥，麻省理工学院电子工程系

Alex Sprintson

美国，德克萨斯州，卡城，德州 A&M 大学。

致　谢

感谢所有作者在书写和校订各章节时所做的贡献、付出的热情和作出的承诺。显然，没有他们的慷慨付出，本书不可能得以完成。此外，还要感谢 Danilo Silva 对各章节的审阅；感谢来自学术出版社的 Tim Pitts、Naomi Robertson、Charlotte Kent 和 Lisa Jones 在本书编辑工作的各个方面所给予的支持、鼓励、理解和帮助。最后，感谢我们的配偶和家庭给予的鼓励和作出的牺牲，他们给了我们进行本项目的空闲时间，也带给了我们灵感。

目　录

第 1 章　网络编码入门

Frank R. Kschischang

加拿大，多伦多，多伦多大学，电子与计算机工程系

摘要

本章是学习网络编码的入门教程，侧重讲述单源多播问题。首先，本章将回顾一些常用的结论，例如：网络的多播容量等于划分该网络中的源节点和目的节点的最小割的容量；通过定义在某足够大的有限域上的线性网络编码，可以实现该多播容量且存在计算该网络编码系数的有效算法；当选取网络编码系数的有限域足够大时，可以无限逼近该多播容量。此外，本章还将讨论实际应用中非常重要的一种网络编码——非相干随机网络编码。在非相干随机网络编码下，源节点和目的节点都没有关于网络中的具体网络编码操作的先验信息。本章的最后将证明对于一般的网络流问题，仅使用线性网络编码并不能实现网络的最大吞吐量，并给出了若干反例。

关键词

图和网络、多播容量、线性网络编码、线性信息流算法、随机线性网络编码、非相干网络编码、非线性网络编码。

1.1　蝶形网络

网络编码建立在一个简单而广泛的概念的基础上：在包交换网络中，中间节点不仅仅是简单地路由转发接收到的数据包，而是对它们进行一些函数操作并计算、转发操作结果。下面将举例说明网络编码带来的好处，第一个例子引自 Ahlswede，Cai，Li 和 Yeung 等人编著的关于网络编码的开创性论文[1]，网络编码这个新兴研究领域也正是起源于此。

图 1-1a 所示的有向图称为“蝶形网络”，它是一个包交换网络模型。在该网络中，源节点 s 需要将信息多播给两个目的节点 t 和 u。有向图的每一条有向边代表一条无差错包传输信道，每个信道利用（Channel Use）可以传输 1 个长为 m 比特的数据包。源节点希望以尽可能高的速率与两个目的节点通信。蝶形网络的一个关键特征是它包含一条瓶颈链路，即图 1-1a 中从节点 b 到节点 c 的有向边（用边 (b, c) 表示）。

解决该多播问题的一种“路由方法”如图 1-1b 所示。源节点在第一个时隙分别给与它相连的两个中间节点发送数据包 x 和 y，网络将 x 转发给目的节点 t 和 u，将 y 转发给目的节点 u。在第二个时隙中，源节点发送数据包 y 和 z，网络将 z 转发给目的节点 t 和 u，将 y 转发给目的节点 t。两个时隙结束后，两个目的节点都能够接收到 x、y、z 这 3 个数据包。使用该路由方法能够在两个时隙内传输给目的节点 3 个不同的数据包，因此该方案的多播吞吐量为 1.5 包/信道利用。这里将网络“信道”的一次“使用”定义

为每条网络边传输最多一个数据包。该多播吞吐量已被证明是使用路由方法能够实现的最大吞吐量。

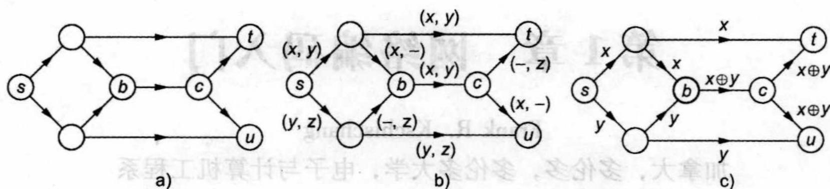


图 1-1

a）包含“瓶颈链路”（b、c）的蝶形网络 b）使用“路由方法”解决多播问题：在2个时隙内将3个数据包x、y、z路由转发给两个目的节点（“—”表示在该时隙内信道空闲） c）使用“网络编码方法”解决多播问题：在瓶颈链路上发送x、y的模二和（异或）$x\oplus y$，使得两个目的节点均能恢复出x和y，实现2数据包/时隙的传输

然而，使用图1-1c所示的“网络编码方法”能够实现2包/信道利用的多播吞吐量。该方法中，第一个时隙源节点仍分发两个数据包x和y，与“路由方法”不同的是，节点b将转发这两个数据包的模二和$x\oplus y$（将x和y按位异或），而不是只转发x、y中的一个。目的节点t接收到数据包x和$x\oplus y$，并根据它们恢复出x和y（$x\oplus(x\oplus y)=y$）。同样地，目的节点u也可以从它接收到的两个数据包中恢复出x和y。网络编码方法以网络中间节点的编码操作和目的节点的解码操作为代价，提升了网络的多播吞吐量，并突破了使用“路由方法”所能实现的吞吐量上限。

那么，我们是否可以做得更好呢？容易看出，对于上述蝶形网络，答案是“不”。节点t和节点u仅通过两条边连接到网络，因此在单位时间内这两个节点最多只能接收两个数据包（一般地，一个网络的最大多播吞吐量取决于分割源节点和目的节点的最小“割集”——引自“最大流最小割”定理，后文将给出详细讨论）。图1-1c中所示的网络编码方法能够实现网络多播吞吐量的最大可能值，因此它是一个最优方案。

另一个例子是在无线中继网络中经常考虑的“无线蝶形网络”。考虑如图1-2所示的网络结构，两个位于山谷的小站s和t不能直接通信，它们需要通过位于山顶的中继节点r进行通信（通过无线信道）。假设该场景中的所有无线设备都只能工作在“半双工”模式，也就是说，1个节点在1个时隙内只能接收或发送（而不能同时收发）1个信号。此外，中继节点r在1个时隙内只可能接收到来自两个山谷小站中其中一个（而不是全部）的信号，但是两个山谷小站都能接收到中继节点广播的信号。假设通信目标是s发送数据包x给t，t发送数据包y给s（数据包的长度均为m比特）。



图1-2 “无线蝶形网络”，山谷小站s和t可通过中继节点r（广播方式）进行通信的无线半双工中继网络

使用“路由方法”可以在 4 个时隙内实现该通信目标：(1) s 发送 x 给 r；(2) r 广播 x 给 t；(3) t 发送 y 给 r；(4) r 广播 y 给 s。

使用“网络编码方法”时，仅需要 3 个时隙就可以实现该通信目标：(1) s 发送 x 给 r；(2) t 发送 y 给 r；(3) r 将 $x\oplus y$ 同时广播给 s 和 x。由于 s 小站本来就有数据包 x，因此它可以通过计算 $x\oplus(x\oplus y)$ 得到数据包 y，相似地，t 也可以恢复出数据包 x。该示例再一次证明了“网络编码方法”以网络中间节点的编码操作和目的节点的解码操作为代价，提升了网络的多播吞吐量，并突破了使用路由方法所能实现的吞吐量上限。

上述两例给我们上了重要的一课：要想充分利用通信网络的信息传输容量，光靠改进路由算法一般来说是不够的。尽管很多教科书在描述包交换网络时都将“数据包流”比喻为“货物流”，但是从蝶形网络示例中可以看出这种比喻从根本上来说就是不合适的。实际上，和在高速公路上行驶的汽车不同，多个信息数据包可以通过多种方式（如“线性叠加”）有效地结合在一起，而目的节点可以从这些结合后的信息中恢复出原始信息。本章乃至于本书都是基于这个简单思想的延拓。

本章结构如下：1.2 节给出了通信网络在多播场景下的一个基本图论模型。1.3.1 节给出了一个从最大流最小割理论推导出的重要定理：网络的最大多播吞吐量等于分割源节点和目的节点的最小“割集”的容量；1.3.2 节详细描述了线性网络编码，该编码方法的本地编码操作只能是定义在有限域上的线性操作；1.3.3 节进一步证明了使用定义在一个充分大的有限域上的线性网络编码可以实现网络的多播吞吐量上限。对于给定的网络，1.4.1 节中给出了能够解决特定多播问题的网络编码系数的有效生成方法；1.4.2 节主要讲述了随机线性网络编码，该编码方法的编码系数随机选取自某域，而不是预先设计好的。当域足够大时，随机选取的编码系数可以获得和预先设计最优的编码系数近似相同的性能。1.5 节中证明了相对于路由方法，编码方法的多播吞吐量增益可以无限大。1.6 节主要讲述的是非相干随机网络编码。使用非相干随机网络编码时，源节点和目的节点都没有关于网络中具体网络编码操作的先验信息。1.7 节中讨论了实现网络编码的字母表所需具备的性质，并给出了若干个关于流问题的示例，这些示例说明（一般情况下）仅使用线性网络编码不足以实现网络的最大传输速率。

网络编码的相关文献日益增多，除本篇之外，其他关于网络编码的入门资料可参考文献 [2-5]。

注意，本章中作如下约定：$|X|$ 表示有限集 X 包含的元素的数量；矩阵用粗体字母表示，如 $\mathbf{A}$；$\mathbf{I}$ 表示单位矩阵；$\mathbf{e}_i$ 表示第 i 列为 1 的单位向量 $(0,\cdots,0,1,0,\cdots,0)$。

1.2　图和网络

1.2.1　组合包网络

下面给出的图论模型虽然不是完全意义上的一般模型，但却足以表征网络编码的主

要思想。

一个组合包网络 $N=(V, E, S, T, A)$ 包括：

1）有限有向无环多重图 $G=(V, E)$，其中 V 表示图 G 的顶点集合，E 表示图 G 的有向边的多重集合。

2）无重复源节点集合 $S \subset V$。

3）无重复目的节点集合 $T \subset V$。

4）有限的数据包符号集 A，$|A| \geqslant 2$。

图 G 中的顶点代表包交换网络中的通信节点，有向边代表通信节点之间的无差错传输信道。有向边 (u, v) 具有单位容量，即每条边每次只能将1个数据包（从符号集 A 中选取的一个符号）从点 u 传送给点 v。如果要进行更大容量（必须为整数）的传输，可以在点 u 和点 v 之间连接若干条平行边，因此 G 是一个多重图。为了避免多条平行边之间相互混淆，我们给多重集合 E 的每条边分配一个标号，分别是 e_1，e_2，…，$e_{|E|}$，每条边都是一组顶点的有序对，且对于 $j \neq i$，允许 $e_i = e_j$。

对于顶点 $v \in V$，我们用 $I(v)$ 表示所有进入点 v 的边的集合，用 $O(v)$ 表示所有从点 v 出发的边的集合。一般来说，对于一个给定的网络，我们可以通过引入“虚拟节点”来实现 $S \cap T = \varnothing$，其中，S 为所有源节点的集合，且满足 $\forall s \in S$，$I(s)=\varnothing$；T 为所有目的节点的集合，且满足 $\forall t \in T$，$O(t)=\varnothing$。

与图论中的设定相同，图 G 中从顶点 $u \in V$ 至顶点 $v \in V$ 的任意（有向）路径都可表示为一个由有向边 (u, v_1)，(v_1, v_2)，(v_2, v_3)，…，(v_{l-2}, v_{l-1})，(v_{l-1}, v) 组成的有限序列，该序列中的每个元素都是有向边集合 E 中的元素。如果一条有向路径的两个端点相同，那么这条路径就被称为图的一个环。包交换网络定义中的“无环”指的是该网络的多重图 G 中不存在环。

对于顶点 u，$v \in V$，如果 $v=u$ 或者图 G 中存在一条从 u 到 v 的有向路径，就称 u 可达 v。上述可达性意味着 V 是部分有序的，通常可以被进一步扩展为兼容的全体有序，又称为拓扑有序。拓扑有序的概念将在1.4.1节中使用到。

在有向边 $(u, v) \in E$ 上进行的操作是将点 u 发出的数据包 $p \in A$ 无差错地交付给点 v。所有边都允许处于空闲状态（即不传输数据包），并假设点 v 具有它的所有入边的状态信息（空闲状态的边可以从 E 中等效移除）。在后文讨论的线性网络编码中，用传输零数据包来表示这些“空闲边”。

组合包网络中的组合逻辑指的是：顶点 v 的出边集合 $O(v)$ 中的任一非空闲边上传输的数据包是来自其入边集合 $I(v)$ 中的所有非空闲边的数据包（当 v 为源节点时，则为它自己产生的数据包）的函数。对于一个实际网络实现，在节点 v 输出的数据包函数被计算好并发送出去之前，进入节点的数据包是需要被缓存的。这里不考虑由传输、缓存以及对数据包的处理引起的时延。

节点 v 使用的函数被称为 v 的本地编码函数。例如，当 v 只有路由功能时，它的出边上发送的数据包可以看做是对入边上接收到的数据包的复制。而网络编码的基本思想是对进入节点的所有数据包进行更一般性的本地编码操作，而不仅仅是简单的复制，从而获得性能上的效益。

对于组合包网络 N，一次信道利用指的是为 E 中的每一条非空闲边都分配一个具体的数据包（取自集合 A），也可以看做是网络中的节点对特定本地编码函数的一次实现。换言之，一次信道利用中的本地编码函数是固定的，且网络中的每条边最多被使用一次。当然，对于不同的信道利用，本地网络编码函数可以不同，图1-1b中给出的“路由方法”可以说明这一点。

本章以在每次信道利用中网络所传输的数据包数量（包/信道利用）作为单位来度量网络的传输效率。如前所述，图1-1b所示的蝶形网络的多播速率为1.5包/信道利用，图1-1c则可达到2包/信道利用。需要注意的是，这里提到的“数据包”所包含的信息量取决于符号集 A 的规模 $|A|$。以包/信道利用为单位度量的速率可以通过乘以 $\log_2|A|$ 比特/数据包将其转化为以比特/信道利用为单位度量的速率。只有在比较使用不同规模的符号集的网络时，如在允许改变符号集规模的条件下，上述两种度量单位才会存在差异。

本节所述的组合网络模型并不具备完全意义上的一般性。在无线传输环境下，需要综合考虑无线信道的广播特性（即一次传输可能会被多个目的节点接收）和无线环境的多址特性（多个节点同时发送时可能会在接收端产生相互干扰），并对该模型进行修正。此外，本模型没有明确考虑中断率（链路不完全可靠的概率）、时延（或其他时序逻辑中关于时间的概念）及可能存在的反馈信道（图论中的环），但使用该组合网络模型足以阐明网络编码的主要思想。

1.2.2　网络信息流

我们用 $S=\{s_1, s_2, \cdots, s_{|S|}\}$ 表示所有源节点的集合。假设每个源节点 $s_i\in S$ 都有无穷多个待发送的数据包，而目的节点 $t\in T$ 只需重建子集 $D_t\subseteq S$ 中的源节点发送的数据包。D_t 表示 t 点播的源节点集合。

假设 s_i 有数据包 p_1，p_2，p_3，…待发送，n 次信道利用后，t（通过对这 n 次信道利用中来自 $I(t)$ 的数据包进行特定的函数运算）可以恢复出 k 个源数据包 p_1，p_2，p_3，…，p_k，那么就说 t 能够以速率 $r_i=k/n$（单位：包/信道利用）解码 s_i 的信息。我们要求源节点 $s_i\in S$ 以同一传输速率 r_i（包/信道利用）服务所有点播它的目的节点。给定网络中的点播集 $\{D_t: t\in T\}$ 以速率 $(r_1, r_2, \cdots, r_{|S|})$ 可达指的是：通过选择本地编码操作（可以是时变的）和 n，所有目的节点都能以给定速率重建它点播的源信息。

关于上述一般网络信息流的一个重要的特殊情况就是多播问题，其中 $\forall t\in T$，$D_t=S$，即每个目的节点都点播了所有源节点的信息。

还有其他一些比较重要的特殊情况，如单点单播（网络中仅有1个源节点 s 和1个目的节点 t，且满足 $D_t=\{s\}$）和多点单播（$S=\{s_1, s_2, \cdots, s_L\}$，$T=\{s_1, s_2, \cdots, s_L\}$ 且满足 $D_{t_i}=\{s_i\}$，L 是大于1的整数）等。

在接下来的章节中，首先将考虑单源多播问题，并证明当数据包符号集是一个足够大的有限域时，使用线性网络编码能够实现最大多播速率。然而我们也将看到，对于一般的网络信息流问题，仅使用线性网络编码通常不足以使网络达到最大速率。

1.3 单源多播问题

1.3.1 多播容量

假设图 G 中点 $s \in V$ 到点 $t \in V$ 是可达的，则一个 $\{s, t\}$ 割指的是一组边的集合 C，该集合满足：从 s 到 t 的所有通路都包含至少一条 C 中的边。也就是说，如果将集合 C 从集合 E 中移除，那么 s 到 t 将不可达。s 到 t 的最小割指的是所有 $\{s, t\}$ 割中权值和最小的割。$\{s, t\}$ 最小割的权值和用 min cut (s, t) 表示。

s 到 t 的最小割可以看做是从 s 到 t 的瓶颈，因为从 s 到 t 的所有通路都需要经过这个最小割。显然，s 到 t 的信息流速率将受到该瓶颈的限制，后面会对这一情况作出详细的说明。

对于任意边集 $B \subseteq E$，B 中元素顺序与 E 中相同，用 X_B 表示 B 中非空闲边上传输的数据包集合。令 s 为源节点，t 为目的节点，C 表示任一 $\{s, t\}$ 割。注意到 $X_{O(s)} \rightarrow X_C \rightarrow X_{I(t)}$ 是一条马尔可夫链。依据信息论中的数据处理不等式（参见参考文献［6］）可知 $I(X_{O(s)}; X_{I(t)}) \leqslant I(X_{O(s)}; X_C)$，即 s 发送的数据包和 t 接收的数据包之间的互信息不大于 s 发送的数据包和通过割集 C 的数据包之间的互信息。考虑到 $I(X_{O(s)}; X_C)$ 的一个明显上界是 $|C|$ 包/信道利用，因此有

$$I(X_{O(s)}; X_{I(t)}) \leqslant |C|$$

其中，C 为任意 $\{s, t\}$ 割。

将所有 $\{s, t\}$ 割中最小的 $|C|$ 代入，可以得到 s 到 t 的信息流速率上界：

$$I(X_{O(s)}; X_{I(t)}) \leqslant \min \operatorname{cut}(s, t)$$

也就是说，如果用 $R(s, t)$ 表示 s 到 t 的通信速率，则有

$$R(s, t) \leqslant \min \operatorname{cut}(s, t)$$

实现这个上界的条件是每一个最小割 C 都满足 $I(X_{O(s)}; X_C) = |C|$，这意味着经过某一最小割的数据包在 A 上独立且均匀分布；此外还要求从任一割集 C（包括 $O(s)$ 在内）传输的数据包中都可以解码得到源数据包。

依据门格尔理论中的边连通定理[7]和商品流理论[8, 9]可知，min cut (s, t) 等于从 s 到 t 的成对的不相交路径的最大数量。上述成对不相交路径的集合可以通过 Ford - Fulkerson 等算法获得。使用路由方法可以实现从源节点 s 到目的节点 t 的速率 $R(s, t) =$ min cut (s, t) 的传输，该路由方法是指每次信道利用都在 s 到 t 的每条不相交路径上传输一个数据包。

接下来我们假设目的节点集合 T 中包含多个（大于 1 个）目的节点，显然，从 s 到集合 T 的多播速率 $R(s, T)$ 小于等于从 s 到任一 T 中的元素 t 的速率，即多播速率 $R(s, T)$ 满足

$$R(s, T) \leqslant \min_{t \in T} \min \operatorname{cut}(s, t) \tag{1-1}$$

关于多播网络的一个重要理论就是使用网络编码可以实现（取等号）式（1-1）中的多播速率上界[1]。此外，当包字符集A是一个足够大的有限域时，使用线性网络编码可以实现该上界[10]。$\min_{t\in T}\min\ \mathrm{cut}(s, t)$被称为该组合包网络的多播容量。

1.3.2　线性网络编码

在正式阐述该理论之前，我们先定义“有限域上的线性网络编码”。在线性网络编码理论中，数据包字符集A是一个包含q个元素的有限域$\mathbb{F}_q$（或者，更一般地，是一个$\mathbb{F}_q$上的向量空间）。在1.10.1节中给出了关于有限域的一些基本知识，不熟悉的读者可以参考。对于本章，读者只需了解有限域提供了一个有限集，在该有限集上明确地定义且有效地实现了加法、减法、乘法及除法运算，并允许系统使用矩阵、行列式、高斯消元法等线性代数中常见的运算工具来解决该域上的联立线性方程组问题。$F_2=\{0, 1\}$是最小的有限域，该域上的所有运算都需要对2取模。

我们用列向量$(p_1, p_2, \cdots, p_r)^{\mathrm{T}}$表示进入源节点$s$的数据包$\boldsymbol{X}_{\mathrm{I}(s)}$。本小节中假设每个数据包$p_i$都是$\mathbb{F}_q$上的一个标量。显然，对于更一般的多个数据包的情况，p_i可以看做是$\mathbb{F}_q$上长度为m的矢量。例如，我们可以用一个定义在$\mathbb{F}_q$上的$r\times m$矩阵来表示$\boldsymbol{X}_{\mathrm{I}(s)}$，该矩阵的第$i$行是$p_i$。1.1节中示例的蝶形网络中传输的数据包定义在$q=2$的有限域上，数据包的长度为$m$比特。

本地编码函数是$\mathbb{F}_q$上的线性函数，即任意中间节点v输出的数据包列向量$\boldsymbol{X}_{\mathrm{O}(v)}$与其接收到的数据包列向量$\boldsymbol{X}_{\mathrm{I}(v)}$之间的关系可以用以下线性方程组表示：

$$\boldsymbol{X}_{\mathrm{O}(v)}=\boldsymbol{L}_v\boldsymbol{X}_{\mathrm{I}(v)}$$

其中，$\boldsymbol{L}_v$是定义在$\mathbb{F}_q$上的系数矩阵，也称为节点v的本地转移矩阵。也就是说，v输出的每个数据包（$\boldsymbol{X}_{\mathrm{O}(v)}$的一个分量）都可以看做是进入$v$的多个数据包（$\boldsymbol{X}_{\mathrm{I}(v)}$的多个分量）在$\mathbb{F}_q$上的线性组合。$\boldsymbol{L}_v$的每一行都对应一条边$e\in O(v)$，称为边$e$的本地编码向量。

需要注意的是，考虑到网络中只允许线性操作，因此任意边上传输的数据包都是源数据包$p_1, p_2, \cdots, p_r$的线性组合。也就是$\forall v\in V$，有

$$\boldsymbol{X}_{\mathrm{I}(v)}=\boldsymbol{G}_v\begin{bmatrix}p_1\\ \vdots\\ p_r\end{bmatrix}$$

其中，$\boldsymbol{G}_v$是定义在$\mathbb{F}_q$上的系数矩阵，也称为v的全局转移矩阵。$\boldsymbol{G}_v$的每一行都对应一条边$e\in I(v)$，称为边e的全局编码向量。

另一种可以有效表示使用线性网络编码的组合包网络的方法是状态－空间描述法。为简单起见，我们从E中移除所有的空闲边并引入网络线图这一概念（参见参考文献[11]及[12]的第168页）。这里用有向图$G'=(V', E')$表示该线图，其中$V'=E$（即G'的顶点对应G的边），当且仅当存在满足$e_i\in I(v)$、$e_j\in O(v)$的点$v\in V$时，$(e_i, e_j)\in E'$才成立。例如，图1-3b中所示的是与图1-3a中的增广蝶形网络对应的线图。

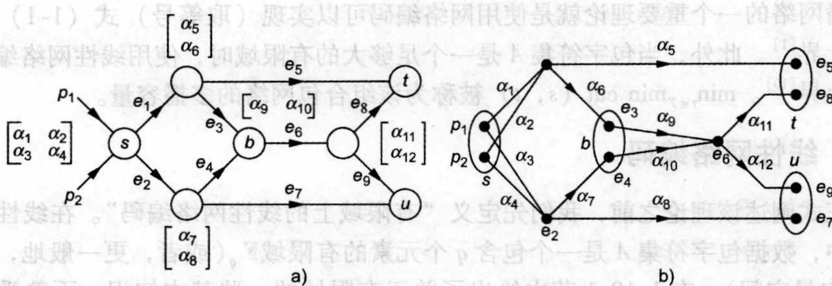


图　1-3

a）标注了本地转移矩阵的蝶形网络　b）与图 a 相对应的线图

假设 $\boldsymbol{X}$ 是一个 $|E|\times 1$ 维的向量，$\boldsymbol{X}$ 的第 i 个分量是 e_i 上传输的数据包，则 $\boldsymbol{X}$ 可以看做是网络的状态。线性组合包网络的一个标准状态－空间模型（在目的节点 t 处）如下：

$$\boldsymbol{X}=\boldsymbol{A}\boldsymbol{X}+\boldsymbol{B}\boldsymbol{U}$$

$$\boldsymbol{X}_{I(t)}=\boldsymbol{C}_t\boldsymbol{X}$$

其中，$|E|\times|E|$ 阶矩阵 $\boldsymbol{A}$ 表示边到边转移矩阵（$\boldsymbol{A}$ 的第 (i,j) 个元素表示从 e_i 到 e_j 的透射系数）；$\boldsymbol{U}=(p_1,\ p_2,\ \cdots,\ p_r)^{\mathrm{T}}$ 表示输入数据包向量；$|E|\times r$ 阶矩阵 $\boldsymbol{B}$ 表示每个输入数据包在 E 中的每条边上的透射系数；$\boldsymbol{C}_t$ 是一个 $|I(t)|\times|E|$ 阶矩阵，它包含的是目的节点 t 关注的 $\boldsymbol{X}$ 的分量。直观上看，矩阵 $\boldsymbol{B}$ 规定了网络中的所有边与进入网络的数据包之间的耦合规则，矩阵 $\boldsymbol{A}$ 规定了网络中的所有边相互之间的耦合规则，矩阵 $\boldsymbol{C}_t$ 则规定了网络中的所有边与目的节点 t 之间的耦合规则。容易看出，t 的全局转移矩阵 $\boldsymbol{G}_t$ 可以表示为

$$\boldsymbol{G}_t=\boldsymbol{C}_t(\boldsymbol{I}-\boldsymbol{A})^{-1}\boldsymbol{B} \tag{1-2}$$

对于任意 k，$\boldsymbol{A}^k$ 中的元素表示 G 中的与 G' 中长度为 k 的路径的两个顶点相对应的两条边之间的耦合系数。考虑到网络是有限且无环的，可知当 L 足够大时，$\boldsymbol{A}^L=0$，即矩阵 $\boldsymbol{A}$ 幂零。因此 $(\boldsymbol{I}-\boldsymbol{A})^{-1}=\boldsymbol{I}+\boldsymbol{A}+\boldsymbol{A}^2+\cdots+\boldsymbol{A}^{L-1}$ 是一个关于 $\boldsymbol{A}$ 的多项式。

下面举一个具体的例子。考虑图 1-3a 中所示的本地转移矩阵，则有 $\boldsymbol{U}=(p_1,\ p_2)^{\mathrm{T}}$，$\boldsymbol{A}=[a_{i,j}]$，其中 $a_{1,3}=\alpha_6$，$a_{1,5}=\alpha_5$（$\forall j\notin\{3,\ 5\}$，$a_{1,j}=0$），$a_{2,4}=\alpha_7$，$a_{2,7}=\alpha_8$（$\forall j\notin\{4,\ 7\}$，$a_{2,j}=0$），$\boldsymbol{A}$ 中的其他元素可依次类推，可以得出

$$\boldsymbol{B}=\begin{bmatrix}\alpha_1 & \alpha_2\\ \alpha_2 & \alpha_4\\ 0 & 0\\ \vdots & \vdots\\ 0 & 0\end{bmatrix},\quad \boldsymbol{C}_t=\begin{bmatrix}0&0&0&0&1&0&0&0&0\\0&0&0&0&0&0&0&1&0\end{bmatrix}$$

$$\boldsymbol{C}_u=\begin{bmatrix}0&0&0&0&0&0&0&0&1\\0&0&0&0&0&0&1&0&0\end{bmatrix}$$

将 p_1 和 p_2 看做输入信号对图 1-3b 进行信号流图处理，或者直接根据式（1-2）可以得到

$$G_t=\begin{bmatrix}\alpha_1\alpha_5 & \alpha_2\alpha_5\\ \alpha_1\alpha_6\alpha_9\alpha_{11}+\alpha_3\alpha_7\alpha_{10}\alpha_{11} & \alpha_2\alpha_6\alpha_9\alpha_{11}+\alpha_4\alpha_7\alpha_{10}\alpha_{11}\end{bmatrix}$$以及

$$G_u=\begin{bmatrix}\alpha_1\alpha_6\alpha_9\alpha_{12}+\alpha_3\alpha_7\alpha_{10}\alpha_{12} & \alpha_2\alpha_6\alpha_9\alpha_{12}+\alpha_4\alpha_7\alpha_{10}\alpha_{12}\\ \alpha_3\alpha_8 & \alpha_4\alpha_8\end{bmatrix}$$

注意到G_t和G_u中的元素均是关于α的多项式，这和前面的分析结果是一致的。

1.3.3　实现多播容量的线性网络编码

对于目的节点t来说，当且仅当$|I(t)|\times r$阶全局转移矩阵G_t的秩为r时，t才能恢复出（p_1，p_2，…，p_r）中的所有元素，这是因为G_t只有满秩时，才存在满足$G_t^{-1}G_t=I_r$的左逆矩阵，其中I_r是$r\times r$阶单位矩阵（注意只要求G_t满秩，而不要求是可以求左逆的方阵）。依据需求，t通过计算$G_t^{-1}X_{I(t)}$可以得到$(p_1, p_2, \cdots, p_r)^{\mathrm{T}}$。对下述定理的证明将说明如果$r\leqslant\min_{t\in T}\min\ \mathrm{cut}(s, t)$且$q$足够大，那么任意目的节点$t$都可以让它的$G_t$成为一个$r\times r$阶可逆矩阵，从而通过线性网络编码实现速率为$r$的多播。

定理1-1（线性网络多播定理）　令$\mathcal{N}=(V, E, \{s\}, T, \mathbb{F}_q)$。当$q$足够大时，使用线性网络编码可以实现速率为$R(s, T)=\min t\in T\min\ \mathrm{cut}(s, t)$的多播。

下面依据参考文献［11］中给出的代数框架来证明上述定理，本书的其他章节也会用到类似的证明方法。其中的一个重要工具称之为“稀疏零点引理”，该引理可以从1.10.2节中证明的引理推导得到。令$\mathbb{F}_q[x_1, \cdots, x_n]$表示由$n$个变量$x_1$，…，$x_n$组成的多项式集合，多项式的系数取自域$\mathbb{F}_q$。

引理1-1（稀疏零点引理）　假设$f\in\mathbb{F}_q[x_1, \cdots, x_n]$是一个不全为0的多项式，对于$\forall i$，$f$的分量$x_i$的次数不大于$d$。那么当$q>d$时，存在$(a_1, \cdots, a_n)\in F_q^n$，满足$f(a_1, \cdots, a_n)\neq 0$。

对定理1-1的证明：令$r=\min_{t\in T}\min\ \mathrm{cut}(s, t)$并用$p_1$，$p_2$，…，$p_r$表示待传输的源数据包。对于任意目的节点$t$，都可以找到$r$条从$s$到$t$的不相交路径。将所有不属于这$r$条路径的边都从$E$中移除（或设置为空闲），我们称得到的仅包含非空闲边的网络为减量网络。减量网络中的所有目的节点$t\in T$都满足$|I(t)|=r$，则所有的全局转移矩阵G_t均为$r\times r$阶矩阵。

依据参考文献［11］，对网络中的所有本地转移矩阵的每一项都引入一个不确定变量，并用α_1，α_2，…，α_η表示，如图1-3a所示。如前所述，G是一个无环网络，因此$G_t(\alpha_1, \alpha_2, \cdots, \alpha_\eta)$（$t\in T$）的所有元素都是多项式形式。我们需要证明的是，通过在一个足够大的有限域上对这些不确定变量的适当选择，可以使得$\forall t\in T$，G_t均可逆。

显然，当且仅当$\det(G_t)\neq 0$时，G_t可逆。实际上，$\det(G_t)$是一个由不确定变量组成的（多元）多项式。考虑到我们至少可以依据路由方法来选择这些不确定变量的取值，即采用单位行列式矩阵$G_t=I_r$，因此该多项式不全为零。

由于网络是无环的，因此一般来说，G_t的任意项中α_i的次数都不大于1。又由于G_t是$r\times r$阶矩阵，因此多项式$\det(G_t)$的最高次数为r，也就是说$\det(G_t)$中的任意

α_i 的次数都不大于 r。而实际上，det（G_t）是一个最高次数为 1 的多元多项式，这是非常令人吃惊的（在 1.10.3 节中给出了相关证明）。

下面我们继续之前的证明，令

$$f(a_1, \cdots, a_\eta) = \prod_{t \in T} \det(\boldsymbol{G}_t(a_1, \cdots, a_\eta))$$

因为 f 等于若干个最高次数为 1 的非零多项式（系数取自某有限域）的乘积，所以 f 是一个最高次数为 $|T|$ 的非零多项式。根据稀疏零点引理可知，只要 $q > |T|$，就可以从 $\mathbb{F}_q$ 中找到满足 $f \neq 0$ 的不确定变量，即满足 $\forall t \in T$，det（$\boldsymbol{G}_t$）$\neq 0$，定理 1-1 得证。

通过在网络中添加一个与每个源节点都连接了若干条平行路径的虚拟“超级源节点”，可以将线性网络多播定理推广到多源（$|S|>1$）多播的情况。具体地说，令 $S = \{s_1, \cdots, s_{|S|}\}$ 并假设 s_i 与目的节点 t 之间有 r（s_i，t）条不相交路径，显然，s_i 能够以速率 r（s_i，t）（单位：包/信道利用）向 t 发送数据包。令 $r_i = \min_{t \in T} r(s_i, t)$ 并令 $r = r_1 + r_2 + \cdots + r_{|S|}$。在网络中添加一个“虚拟节点”$s^*$，该节点满足：$s^*$ 与 s_i 之间连接了 r_i 条平行边。注意到 $\forall t \in T$，s^* 与 t 之间都存在 r 条不相交路径。根据定理 1-1 可知，通过使用网络编码，s^* 能够以 r 包/信道利用的速率与所有目的节点进行多播通信，而我们要做的也正是寻找这样一种网络编码方案。实际上，存在 $r \times r$ 阶矩阵 $\boldsymbol{L}_{s^*}$，满足

$$\boldsymbol{X}_{O(s^*)} = \boldsymbol{L}_{s^*}(p_1, p_2, \cdots, p_r)^{\mathrm{T}}$$

即 s^* 的出边上传输的数据包是 r 个超级源数据包的线性组合。然而，由于 O（s^*）是将网络划分为 s^* 和除 s^* 外其余部分的一个割，因此我们一定可以通过解码 $X_{O(s^*)}$ 得到（p_1，p_2，$\cdots$，p_r），这意味着矩阵 $\boldsymbol{L}_{s^*}$ 必须可逆。对 s^* 预乘矩阵 $\boldsymbol{L}_{s^*}^{-1}$ 后，上述一般性网络编码方案就变成一个 O（s^*）的所有边上的全局编码向量都是单位向量的网络编码方案，且不会影响目的节点的解码能力。此时，源节点 s_i 将接收到 r_i 个“未编码的”超级源数据包。如果把这 r_i 个包看做是在 s_i 上产生的数据包，则上述线性网络编码方案可以实现 s_i 与所有目的节点之间速率为 r_i（单位：包/信道利用）的数据包传输，进而实现源节点组 $\{s_1, \cdots, s_{|S|}\}$ 与目的节点之间速率为（r_1，$\cdots$，$r_{|S|}$）的传输。

定理 1-1 还有另一个有用的推论。令 $\mathcal{N} = (V, E, \{s\}, T, \mathbb{F}_q)$，当 q 足够大时，存在满足 $\forall t \in T$，$\boldsymbol{G}_t$ 的秩为 min cut（s，t）的线性网络编码方案。也就是说，所有目的节点的全局转移矩阵可以同时达到它们的最大可能秩。要证明上述推论，令 $r = \max_{t \in T}$min cut（s，t），并对每个目的节点 $t \in T$ 都添加 r − min cut（s，t）条从 s 到 t 的平行边。根据定理 1-1，添加平行边后的网络可以使用线性网络编码方案实现速率为 r 的多播传输，该编码方案在每个目的节点 $t \in T$ 处都引入一个秩为 r 的全局转移矩阵。如果把添加的平行边从 t 中移除，则全局转移矩阵中与这些边对应的行也会被移除，得到的 $\boldsymbol{G}_t$ 的秩即为 min cut（s，t）。

1.4 多播网络中的码字构造

1.4.1 线性信息流算法

定理 1-1 说明当 $q > |T|$ 时，使用有限域 $\mathbb{F}_q$ 上的线性网络编码方案可以实现网络

的单源多播容量。给定一个网络，我们能否有效地找到相应的线性网络编码方案？幸运的是，答案是“可以”。本节将简单介绍参考文献［13］中的“线性信息流”算法，对于上述问题，该算法给出了一个多项式时间解决方案。和对定理 1-1 的证明相同，令 $r=\max_{t\in T}\min\ \mathrm{cut}\ (s,\ t)$，并用 $(p_1,\ p_2,\ \cdots,\ p_r)$ 表示需要传输的源数据包。假设 $|T|\geqslant 2$，并且给定一个满足 $q\geqslant |T|$ 的有限域 $\mathbb{F}_q$。注意，此处可取等号 $q=|T|$。线性信息流算法可以分为两个阶段。

在算法的第一阶段，我们使用最大流量算法（Ford – Fulkerson 算法）找到 r 条从 s 到 t 的不相交路径，分别用 $p_1(t)$，…，$p_r(t)$ 表示。我们只需要考虑这 r 条路径中的边，因此将不属于这些路径的边从 E 中移除（或置为空闲），并称所得到的没有空闲边的网络为减量网络。对于该减量网络中的边 e、目的节点 $t\in T$ 以及标号 $i\in\{1,\ \cdots,\ r\}$，定义如下指示函数：

$$\delta(e,\ t,\ i)=\begin{cases}1, & e\in p_i(t)\\ 0, & \text{其他}\end{cases}$$

该函数指示了 e 是否属于从 s 到 t 的第 i 条路径 $p_i(t)$。对于给定的 e 和 t，由于从 s 到 t 的 r 条路径不相交，因此等式 $\delta(e,\ t,\ i)=1$ 最多只对一个 i 值成立。如果存在满足 $\delta(e,\ t,\ i)=1$ 的 i，就说 e 可以影响 t。

在算法的第二阶段，我们使用以下方法为减量网络中的每条边都选择一个 $1\times r$ 阶全局编码向量。对边的处理依照其拓扑顺序进行，也就是说，如果点 v_i 拓扑优先于点 v_j，那么就先处理 $O(v_i)$ 中的边，再处理 $O(v_j)$ 中的边。这样可以保证给定边的前趋边（即“上游的边”）优先选择编码向量。具体地说，如果有向路径 $p_{i(t)}$ 依次由边 e_{i_1}，e_{i_2}，…，e_{i_L} 组成，则在选择全局编码向量时，e_{i_1} 先于 e_{i_2}，e_{i_2} 先于 e_{i_3}，依此类推。因此，在该算法的任意时刻，每条路径 $p_i(t)$ 中都存在一条最近更新边 e_{i_j}，满足：我们已经为 e_{i_j} 分配了全局编码向量，但还没有为与其相邻的 $e_{i_{j+1}}$ 分配全局编码向量。用 $\boldsymbol{h}_i(t)$ 表示分配给 $p_i(t)$ 的最近更新边的全局编码向量。在初始化后，该算法将保持一个核心的不变性，即对于任意目的节点 $t\in T$，$\{\boldsymbol{h}_1(t),\ \cdots,\ \boldsymbol{h}_r(t)\}$ 都是 $\mathbb{F}_q^r$ 的一个基，也就是说 $r\times r$ 阶矩阵

$$\boldsymbol{H}_t=\begin{bmatrix}\boldsymbol{h}_1(t)\\ \vdots\\ \boldsymbol{h}_r(t)\end{bmatrix}$$

非奇异。该算法结束时，目的节点 t 的最近更新边应为其入边集合 $I(t)$，矩阵 $\boldsymbol{H}_t$ 则应等于其全局转移矩阵 $\boldsymbol{G}_t$。

在算法的第二阶段进行的初始化实际上是为所有 $O(s)$ 中的边选择全局编码向量。每个目的节点 t 上都保持着一个由 $p_1(t)$，…，$p_r(t)$ 的最近更新全局编码向量组成的集合 H_t（初始值为空）。当为边 $e\in O(s)$ 选择全局编码向量 $\boldsymbol{h}(e)$ 时，需保证 $\boldsymbol{h}(e)$ 与所有 $\boldsymbol{H}_t$ 中的向量线性独立，这里的 t 指的是所有被 e 影响的目的节点（即使得 $\delta(e,\ t,\ i)=1$ 成立的所有 $t\in T$，其中 $i\in\{1,\ \cdots,\ r\}$）。考虑到 $|H_t|<r$（这是因为 e 影响 t 就意味着 H_t 中已分配的向量数量小于 r），并假设所有 H_t 都不选取零向量，则

$\boldsymbol{h}(e)$最多能够从$q^{r-1}-1$个非零向量中取值。在$\mathbb{F}_q^r$包含的q^r个向量中，$\boldsymbol{h}(e)$不能选择的向量数量最多为$1+|T|(q^{r-1}-1)$，也就是说可以选择的向量数量最少为

$$q^{r-1}(q-|T|)+|T|-1$$

当$q\geqslant|T|$时，该值严格正，因此上述初始化过程可以保证对全局编码向量的分配满足：每个集合H_t都是$\mathbb{F}_q^r$的一个基，即矩阵$\boldsymbol{H}_t$非奇异。

初始化完成后，接下来考虑对边e的处理，如图1-4所示。对e分配本地编码向量将使相应的H_t中的一个全局编码向量被替换，如图1-4中的阴影框所示。所有阴影框都将被同一个向量替换，该向量是图中所有进入点v的全局编码向量的线性和。假设e的所有入边上的全局编码向量张成一个m维向量空间V_1。目的节点t上的所有未被替换的向量张成一个$(r-1)$维向量空间V_2，则空间$V_1\cap V_2$的维度为

$$\begin{aligned}\dim(V_1\cap V_2)&=\dim(V_1+V_2)-\dim(V_1)-\dim(V_2)\\&=r-m-(r-1)=m-1\end{aligned}$$

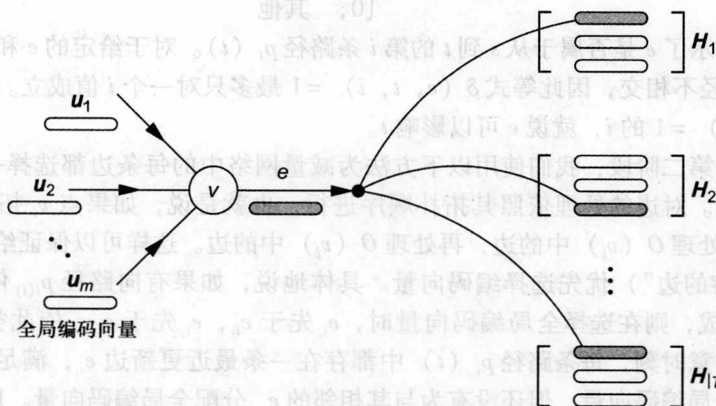


图1-4 线性信息流算法的关键步骤，边e上的编码向量需为“进入的”全局编码向量u_1，…，u_m的线性和，这样才能保证所有目的节点$t\in T$上的矩阵$\boldsymbol{H}_t$可逆

考虑到V_1肯定包含用于替换的向量，故$\dim(V_1+V_2)=r$。空间$V_1\cap V_2$由一个零向量和$q^{m-1}-1$个非零向量组成，为e选取全局编码向量时，应将$V_1\cap V_2$中的所有向量都排除在可选范围之外。总地来说，在V_1包含的q^m个向量中，不能选择的向量数量最多为$1+|T|(q^{m-1}-1)$，也就是说，可以选择的向量数量最少为

$$q^{m-1}(q-|T|)+|T|-1$$

当$q\geqslant|T|$时，该值严格正，从而确保该算法成功。

需要注意的是，如果简单地从V_1中随机选取e的本地编码向量，假设前一次选取是成功的，则本次选取成功的概率至少是$1-|T|/q$，这里的成功是指所选取的向量能够使所有矩阵$\boldsymbol{H}_t$都维持之前的秩。我们可以通过从$\mathbb{F}_q$中独立均匀地随机选取本地编码系数来实现对上述编码向量的随机选择。

参考文献[13]中证明了线性信息流算法的运行时间可以表示为关于$|E|$、$|T|$

以及 log（q）的多项式，因此我们可以有效地实现该算法。该有效性的重要来源之一在于：在（$r-1$）维向量空间中检验长度为 r 的向量的隶属关系可以等效为在（r，$r-1$）线性码中检验该向量的隶属关系；而上述（r，$r-1$）线性码可以通过一致校验方程来描述（也就是说该对偶码是一维的），因此实现该检验只需要进行一次求内积运算。

Harvey 等人在参考文献［14］中提出了一种能够替代线性信息流算法的矩阵实现法。Barbero 和 Ytrehus 在参考文献［15］中给出了实现较小域规模的方法。

1.4.2 随机构造

假设我们不像线性信息流算法一样对线性网络码进行详尽的设计，而是简单地完全随机地选取本地编码向量，那么所得到的线性网络编码能够实现网络多播容量的概率是多大呢？Ho 等人在参考文献［16］中对上述“随机方法”进行了研究，在实际应用中，随机线性网络编码带来了很多好处，我们将在第1.6节详细讨论。

令 $\eta \leqslant |E|$ 表示“随机编码边”（随机选取本地编码向量的边）的数量，考虑1.4.1节中描述的线性信息流算法操作。假设该算法每次操作一条随机编码边时，都从 $\mathbb{F}_q$ 中独立均匀地随机选取该边的本地编码系数（同时还假设对非随机编码边上的本地编码系数的选取可以保证所有矩阵 $\boldsymbol{H}_t$ 满秩）。如前文所述，对第一条随机编码边操作成功（即相应的全局编码向量使得所有矩阵 $\boldsymbol{H}_t$ 满秩）的概率至少是 $1-|T|/q$；在第一次操作成功的条件下，对第二条随机编码边操作成功的概率仍至少是 $1-|T|/q$；因此两次操作都成功的概率至少是 $(1-|T|/q)^2$。依次类推，当之前的操作都成功时，第 i 次操作成功的概率至少是 $1-|T|/q$，η 次操作全部成功的概率为

$$P[\text{success}] \geqslant \left(1-\frac{|T|}{q}\right)^{\eta} \geqslant 1-\frac{\eta|T|}{q}$$

要使 P［success］$\geqslant 1-\delta$（δ 为正且很小）需满足

$$q \geqslant \frac{|T|\eta}{\delta}$$

由此可以看出，当域规模 q 足够大时，随机选取的网络编码能够以极大的概率实现多播容量。实际上，q 最好是2的 m 次幂，其中 m 是域元素的比特长度。我们还可以发现，随机线性网络编码不能实现多播容量的概率 P［failure］满足

$$P[\text{failure}] \leqslant |T|\eta 2^{-m}$$

该概率随 m 的增大而指数减小。

参考文献［16］中的分析进一步证明了这个结果。对于某些情况（如 $|I(v)|=1$ 时），节点 v 可以为它的某条出边找到肯定不会降低可达多播速率的本地编码向量。如果一条边的本地编码向量不是一个固定的确定值而是随机选取得到的，那么就称这条边为“随机编码边”。对于线性信息流算法的第一阶段，令 c_t 表示从 s 到 t 的 r 条不相交路径所包含的随机编码边的数量，并令 $\eta' = \max_{t\in T} c_t$，则随机选择的本地编码系数构成一组可行的网络码的概率 P［success］满足

$$P[\text{success}] \geqslant \left(1-\frac{|T|}{q}\right)^{\eta'}$$

证明这个下界的关键是使用改进的多项式符号检验定理（Schwartz - Zippel Theorem），详见 1.10.2 节。

1.5　编码和路由

下面将举例证明编码相对于路由的增益可以无限大（参见参考文献［13］）。考虑图 1-5 所示的组合包网络，网络中包含了 1 个源节点 s，$2h$ 个中间节点 u_1，…，u_{2h}，以及 $\binom{2h}{h}$ 个目的节点 t_1，…，$t_{\binom{2h}{h}}$。每个目的节点都连接到不同的包含 h 个中间节点的子集上。

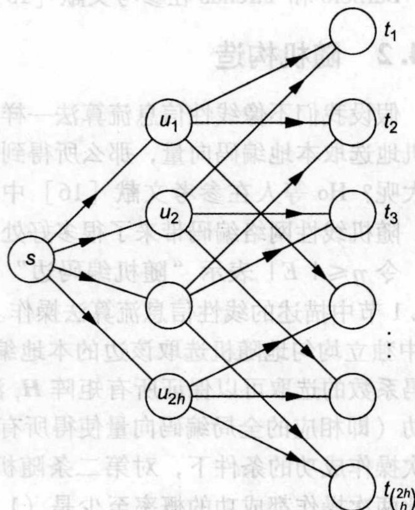


图 1-5　由 $2h$ 个中间节点以及 $\binom{2h}{h}$ 个目的的节点组成的网络，每个目的节点都连接到不同的中间节点子集上。该网络的多播容量为 h，仅使用路由方法可实现的多播容量小于 2

该网络的多播容量是 h（单位：包/信道利用）。使用经典的长度为 $2h$、线性速率为 1/2 的最大距离分割容错码 C 可以实现该容量，如$\mathbb{F}_q$上的截断式里德所罗门码（Reed - Solomon Code），其中 $q \geqslant 2h$（参见参考文献［17］等）。这类码可以纠正长度为 $2h$ 的符号组中所有长度为 h 的删除图样。本例中，s 将第 i 个码字符号（是 h 个源符号的一个线性组合）发送给中间节点 u_i，u_i 将该符号简单地转发给与它连接的所有目的节点。每个目的节点都能收到 $2h$ 个码字中的 h 个，因此可以通过使用与 C 相对应的删除 - 纠正算法进行解码。实际上，本例中的所有的“网络编码”都发生在网络的源节点而不是中间节点上（指对源数据包求线性组合）。

使用路由方法时，多播容量小于 2 包/信道利用且与 h 的取值无关。为了说明这个问题，假设源节点需要以 2 包/信道利用为速率通过 n 次信道利用将 $2n$ 个数据包 p_1，p_2，...，p_{2n}路由转发给所有目的节点，令 U_i 表示所有接收到 p_i 的中间节点组成的子集。由于发送给所有中间节点的数据包的总数量最大为 $2nh$，因此必然有

$$\sum_{i=1}^{2n} \mid U_i \mid \leqslant 2nh$$

且必然存在 j，满足 $\mid U_i \mid \leqslant h$（根据鸽巢原理）。也就是说，$p_j$ 最多被发送给 h 个中间节点，因此至少有 h 个中间节点没有收到 p_j，用 W 表示这些节点组成的集合。至少存在一个目的节点 t_W 满足：所有与 t_W 相连的中间节点都属于 W，那么 t_W 将接收不到 p_j。因此，使用路由方法实现的多播速率小于 2 包/信道利用。

使用与给定网络的最优多播树排样对应的（可分的）路由方法一般可以实现该网络的最大多播容量，我们称之为“可分的斯坦纳（Steiner）树 - 排样数”。G 的一个多播树

是指 G 中的一个贯穿源节点和所有目的节点的有向子树。令 K 表示所有多播树的集合，K_e 表示 K 中所有经过链路 e 的多播树组成的一个子集。对于 K 中的树 $k \in K$，我们给它分配大小为 f_k 的流速，即 k 包含的每条边的流速都为 f_k，那么可实现的最大速率为

$$\max_{f_{k, k \in K}} \sum_{k \in K} f_k$$

$$\text{其中：} \forall e \in E, \sum_{k \in K_e} f_k \leqslant 1$$

$$\forall k \in K, f_k \geqslant 0$$

上述线性规划要求对每个树 $k \in K$ 的流分配能使总流最大，且遵循以下约束条件：任意（单位容量的）边 $e \in E$ 上分配的流量不超过单位流量；所有的流量都是非负的。

考虑到 G 中的多播树的数量可能会很大，因此除非网络规模很小，否则求解上述线性规划在计算上通常是不可行的。实际上，寻找分布树的最优可分排样是一个 NP 困难问题[18]，但我们却可以通过某种计算有效的方法得到能够实现多播容量的网络码。

参考文献［19］中证明了多播码的增益等于最小权值有向斯坦纳树问题的线性规划松弛方法的整性间隙，该增益是指使用网络编码实现的多播速率与使用路由方法实现的多播速率的比值。已知该整性间隙的下界是 $\Omega\left((\log|V|/\log\log|V|)^2\right)$[20] 和 $\Omega\left(\sqrt{|V|}\right)$[21]。Chekuri 等人将参考文献［21］中给出的工具引入参考文献［22］中，构建出的网络多播码的增益与 $\sqrt{|T|}$ 成正比，图 1-5 中的多播码增益为 $\Omega(\log|T|)$。

1.6　非相干网络编码

1.6.1　携带数据包报头的传输

除了用于设计特定的网络编码以外，随机线性网络编码方法还可以用于网络协议的设计[16, 23]。

使用该方法时，源节点和目的节点之间的通信是以轮次或者“代”为单位进行的。每一“代”通信中，源节点会向网络中注入若干长度固定的数据包，每个数据包都可以看做是有限域 $\mathbb{F}_q$ 上的长度固定的行向量。这些数据包在网络中传播时有可能会经过一些发射机和接收机之间的中间节点。当中间节点有机会发送数据包时，它将在 $\mathbb{F}_q$ 上构建一个关于该节点上的所有可用数据包的随机线性组合，并发送该组合。目的节点则需要收集这些随机产生的数据包，并尝试恢复出注入网络的数据包集。即使我们不做任何关于网络运行同步、无延迟甚至无环的假设，该模型仍具有鲁棒性。

根据某代通信中成功传输的数据包集合可以构建相应的有向多重图，该多重图的顶点集与原网络相同，多重图的边表示成功的数据包传输。正如我们看到的，源节点与目的节点之间的信息传输速率（包/代）的上界等于它们之间的最小割，且当 $q \to \infty$ 时，$\mathbb{F}_q$ 上的随机线性网络编码能够以趋近于 1 的概率实现该最小割速率[16]。

令 $\boldsymbol{P}$ 表示将源数据包作为行向量所构成的矩阵，则有

$$\boldsymbol{P}=\begin{bmatrix} p_1 \\ \vdots \\ p_r \end{bmatrix}$$

假设目的节点 t 已经收集了该代中的 L 个编码数据包 y_1，…，y_L，并令 $\boldsymbol{Y}$ 表示将这些数据包作为行向量所构成的矩阵，则有

$$\boldsymbol{Y}=\boldsymbol{G}_t\boldsymbol{P}$$

其中，$L\times r$ 阶矩阵 $\boldsymbol{G}_t$ 表示通过随机选择编码系数构建的目的节点 t 上的全局转移矩阵。当源节点和目的节点都未知 $\boldsymbol{G}_t$ 时，相应的网络操作称为非相干网络编码。

那么目的节点是如何恢复出 $\boldsymbol{P}$ 的呢？对于非相干多天线通信，一个显而易见的方法是给发送的数据包添加所谓的“报头”，这些报头记录了所有接收数据包的各个分量的具体线性组合情况。换句话说，令 $p_i=(\boldsymbol{e}_i,\ \boldsymbol{m}_i)$，其中 $\boldsymbol{e}_i$ 是第 i 个分量为 1 的单位向量，$\boldsymbol{m}_i$ 是“有效载荷”，则 $\boldsymbol{P}=[\boldsymbol{I}_r\quad \boldsymbol{M}]$，那么

$$\boldsymbol{Y}=\boldsymbol{G}_t\boldsymbol{P}=\boldsymbol{G}_t[\boldsymbol{I}_r\quad \boldsymbol{M}]=[\boldsymbol{G}_t\quad \boldsymbol{G}_t\boldsymbol{M}]$$

$\boldsymbol{Y}$ 的前 r 列是矩阵 $\boldsymbol{G}_t$，当 $\boldsymbol{G}_t$ 的左逆 $\boldsymbol{G}_t^{-1}$ 存在时（当域足够大时，$\boldsymbol{G}_t^{-1}$ 存在的概率很大），t 可以恢复出 $\boldsymbol{M}=\boldsymbol{G}_t^{-1}(\boldsymbol{G}_t\boldsymbol{M})$。

在固定网络中通信时，目的节点可以以先验信息的形式预先获得全局编码向量，相比较而言，使用数据包报头会带来传输开销；然而在实际的传输方案中，数据包的长度通常很大，报头开销可能只占传输信息总量的很小一部分。例如，取 $q=2^8=256$（即域元素的单位是字节），分代大小 $r=50$，数据包长度为 $2^{11}=2048$ 字节，此时报头开销为 50 字节，仅占整个数据包的 $50/2048=2.4\%$。

1.6.2 子空间传输

那么在使用非相干随机线性网络编码信道通信时，这种传输数据包报头的方法是否必要呢？网络的结构会影响 $\boldsymbol{G}_t$ 的构造，要想对该影响具有鲁棒性，有一种方法是随机构造 $\boldsymbol{G}_t$ 并使 $\boldsymbol{G}_t$ 以较大的概率满秩。当 $\boldsymbol{G}_t$ 是一个秩为 r（r 为每代数据包的数量）的随机矩阵时，$\boldsymbol{Y}=\boldsymbol{G}_t\boldsymbol{P}$ 还保留有 $\boldsymbol{P}$ 的哪些特性呢？由于 $\boldsymbol{G}_t$ 是随机的，因此由乘积 $\boldsymbol{G}_t\boldsymbol{P}$ 唯一可以确定的是 p 的行空间。实际上，就目的节点 t 而言，该空间的所有生成集都是等价的。

根据上述内容，参考文献［24］中提出了一个有趣的在非相干随机网络编码信道中传输信息的方法。该方法从 $\boldsymbol{P}$ 的各个行向量张成的向量空间中选择要发送的信息。即使 $\boldsymbol{G}_t$ 不满秩，目的节点接收到的也是 $\boldsymbol{P}$ 的行空间的子空间；如果能够设计一个（关于向量空间的）码本，使得码本中的每个码字只包含一个这样的低维度子空间，那么就可以纠正这种维度的“丢失”。当网络中存在信道噪声或恶意攻击导致注入网络的数据包不是源节点发送的数据包时，甚至可以通过设计码本来纠正这种“错误”。

参考文献［24］、［25］等给出了上述用于在非相干随机网络编码信道中传输信息的子空间方法。研究人员已经构建出了一系列与里德所罗门码性质相似的错误-删除纠正码；我们可以把这一类码看做是在非相干网络编码信道下对经典训练矩阵度量码[26]

的改进方案。这类方案可以提供对于错误和删除的鲁棒性，也可以对抗窃听者对低秩的信息向量线性组合的窃听。

1.7　关于字母表和非线性

前面提到，要实现网络的多播容量，需要在足够大的有限域上进行线性网络编码，而实际上，当有限域的大小满足 $q > |T|$ 时就已足够使用了。

假设我们希望最小化字母表的大小。参考文献［28］中的研究表明，对于给定的多播网络问题，仅仅是判断 $\mathbb{F}_q$ 上的线性网络编码是否存在都是一个 NP 困难问题。

下面我们将参考 Riis 和 Ahlswede 的研究[29]来举例说明为什么这类关于网络编码字母表大小的问题会如此困难。考虑图 1-6 中的网络，源节点 s 要发送 2 个信息 x 和 y（假设取自有限集 A）给 5 个目的节点，所有目的节点都不能单独接收到 x 和 y，但可以接收到 x、y、$x \oplus y$ 和 $x \boxplus y$ 的各种组合。$\oplus$和$\boxplus$是将 $A \times A$ 映射到 A 的函数，也可以将它们看做是某种二进制运算符。

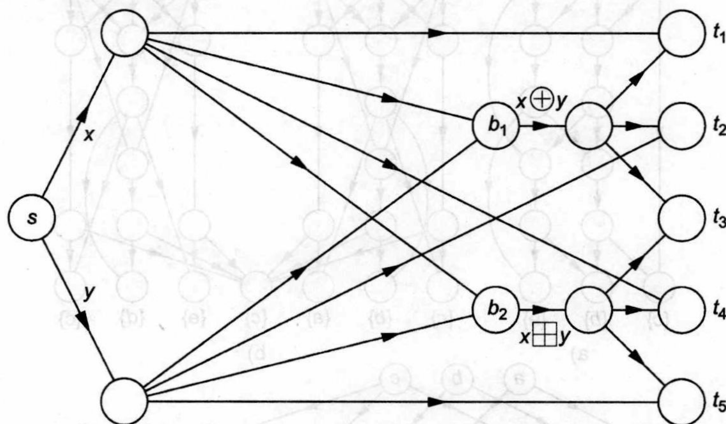


图 1-6　$\oplus$和$\boxplus$的运算表必须构成一对正交拉丁方阵，目的节点才能够根据 x、y、$x \oplus y$ 和 $x \boxplus y$ 的各种组合恢复出 x 和 y，因此当字母表的大小等于 2 或者 6 时，图示多播问题无解

那么，$\oplus$和$\boxplus$运算符应该具备哪些性质呢？显然我们希望能够根据 $x \oplus y$ 和 y 恢复出 x，或是根据 $x \oplus y$ 和 x 恢复出 y，因此$\oplus$运算符的运算表必须是一个拉丁方阵㊀。类似地，$\boxplus$运算符的运算表也必须是一个拉丁方阵。实际上，由于 t_3 想要根据 $x \oplus y$ 和 $x \boxplus y$ 恢复出 x 和 y，因此这两个运算表必须构成所谓的正交拉丁方阵，即通过 $x \oplus y$ 和 $x \boxplus y$ 的值能够唯一确定 x 和 y。因此通过字母表 A 上的网络编码是否可以解决图 1-6 中的多播问题与是否存在秩为 $|A|$ 的一对正交拉丁方阵这两个问题是等价的。

成对的正交拉丁矩阵何时存在的问题可以追溯到欧拉时代，但直到 1959 年才得到

㊀ 秩为 m 的拉丁方阵是指一个 $m \times m$ 阶矩阵，它满足：m 个不同的元素中的每一个都在矩阵的每行或者每列中出现且仅出现一次。1.10.1 节中 F_4 上的加法表就是一个 4×4 拉丁方阵的例子。

彻底的解决（参阅参考文献［29］中的相关讨论）。已有的研究证明，除了 $m=2$ 和 $m=6$ 以外，其余阶数的正交拉丁方阵都存在。因此只要所使用的字母表大小不等于2或者6，图1-6中的多播问题就存在解决方案。注意，字母表大小等于3时存在解决方案并不意味着对于所有大于3的字母表都存在解决方案。

对于非多播网络中的通信问题，由于不同网络流之间的潜在“干扰”，情况将变得更加复杂，此时仅使用线性网络编码并不足以解决这样的网络信息流问题。

举例说明，如图1-7a所示的网络中有3个源节点 a、b 和 c，图中标示出了每个目的节点点播的内容。参考文献［30］中Dougherty，Freiling和Zeger等人证明了该网络仅容许特征值为2的有限域上的标量线性解决方案。另一方面，图1-7b中的网络仅容许特征值不为2的有限域上的标量线性解决方案。图1-7c中的网络是这两个网络的结合，因此任意域上的线性方案对于该网络都是不可行的（对于图1-7c中的通信问题，参考文献［30］给出了一个字母表大小等于4的非线性解决方案）。这个“可怕的”的例子令人信服地证明了线性网络编码并不足以解决所有的网络编码问题。

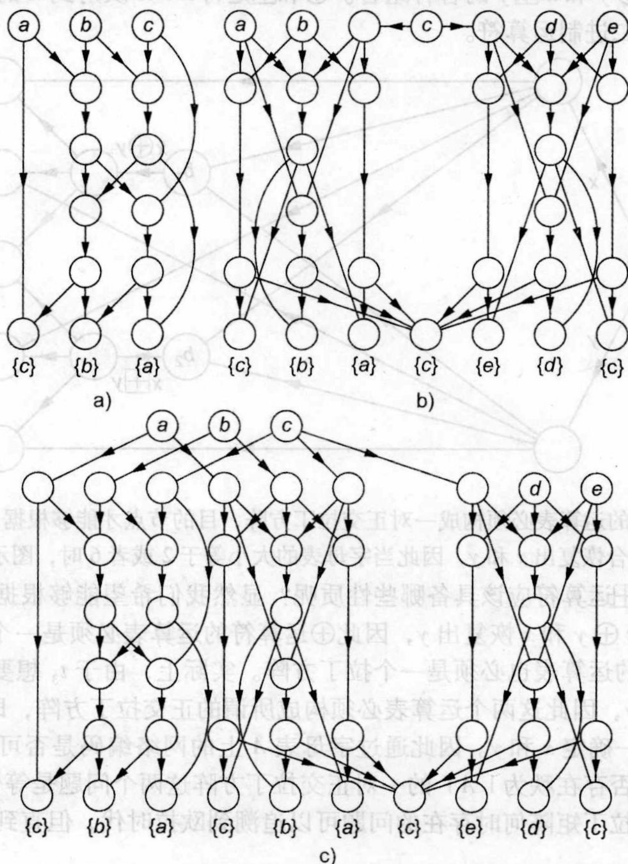


图1-7　一个“可怕的”网络示例。根据图中所示的网络结构和点播内容

a）网络仅在特征值为2的有限域上存在标量线性解决方案　b）网络仅在特征值不为2的有限域上存在标量线性解决方案

c）网络是a网络和b网络的结合，因此对于c网络不存在线性解决方案，而只有一个字母表大小等于4的非线性解决方案

1.8 结论

正如本章中所描述的，网络编码的思想既简单又深远。网络编码允许网络中的节点进行编码（而不仅仅是路由），使用线性网络编码可以实现网络的多播容量，与路由方法相比，网络编码的潜在的吞吐量优势（在某些情况下）是无穷大的。对于给定的多播问题，最优线性网络编码方法可以在多项式时间内找到（或者是简单地随机获取，这种方式成功的概率很大），而寻找最优路由方法却是一个 NP 困难问题。在实际应用中，从随机线性网络编码方法可以拓展出对网络底层拓扑结构变化具有鲁棒性的多播协议。本章还给出了一个精心设计的“可怕的”例子，这个例子证明了对于一般的多播流问题，仅使用线性网络编码不足以实现其最优吞吐量。

本书后面的章节探讨了网络编码在各种场景中的应用，祝读者阅读愉快！

1.9 致谢

十分感谢 Chen Feng、Ashish Khisti、Danilo Silva 和 Alex Sprintson 对本章节内容的早期版本提出的有益建议。对 1.10.3 节中的引理 1-3 的证明由 Joachim Rosenthal 完成。

1.10 附录

1.10.1 有限域

这里只对有限域作一个简单的介绍。从代数角度看，域是指一个集合以及该集合上的两种明确定义的二进制运算（加法运算 + 及乘法运算 ×），这两种运算遵循大家所熟知的运算规则。

域上的加法运算必须满足结合律和交换律；一个域必须包含一个加法恒等元（用 0 表示）；域中的每个元素 x 必有一个加法逆元（用 $-x$ 表示）。也就是说域中的元素可对加法构成阿贝尔群。

域上的乘法运算必须满足结合律和交换律；一个域必须包含一个乘法恒等元（用 1 表示）；域中的每个元素 x（0 除外）必有一个乘法逆元（用 x^{-1} 表示）。也就是说，域中的所有非零元素可对乘法构成阿贝尔群。

此外，域上的乘法运算还应满足对加法的分配律，即对于域中的任意元素 x、y、z，均有 $x\times(y+z) = (x\times y) + (x\times z)$。

域上的减法运算和除法运算是通过逆元来定义的，即 $x-y=x+(-y)$，$x/y=x\times y^{-1}$，后者仅当 $y\neq 0$ 时成立。乘号一般都省略不写，例如，假设 x 和 y 是域中的元素，则可以用 xy 表示 $x\times y$。对于任意正整数 a 和域元素 x，有如下表示：

$$x^a = \underbrace{x \times x \times \cdots \times x}_{a\text{次}}$$

关于（有限）域的几个大家熟悉的例子有：有理数集合、实数集合和复数集合，以及这些集合上相应的加法运算和乘法运算。由于整数集合中除了 ±1 以外，没有任何非零整数的乘法逆元仍是整数，因此整数集合不能构成一个域。如果 p 是一个素数，则集合 $\{0, 1, \cdots, p-1\}$ 上的模 q 的整数加法运算和整数乘法运算是封闭的，因此该集合可构成一个域，用 $\mathbb{F}_q$ 表示。

下面举例说明，最小的有限域仅包含两个元素：0 和 1，用 $\mathbb{F}_2$ 表示，$\mathbb{F}_2$ 上的加法和乘法运算表（即对 2 取模的整数加法运算和整数乘法运算）如下所示：

+	0	1	×	0	1
0	0	1	0	0	0
1	1	0	1	0	1

它们分别与逻辑或（XOR）和逻辑与运算相对应。需要注意的是，在 $\mathbb{F}_2$ 上 $-1=1$。

有一些从域公理推出的结论对于所有的域 $\mathbb{F}$ 都适用。例如，$\mathbb{F}$ 上的任意元素 x 都满足：$0 \times x = 0$，$-1 \times x = -x$；当且仅当 x 和 y 都不为零时，两个元素的乘积 $x \times y$ 不为零；当且仅当 x 和 y 中至少有一个是零时，两个元素的乘积 $x \times y$ 等于零。

考虑定义在有限域 $\mathbb{F}$ 上的序列 1，1+1，1+1+1，…，由于 $\mathbb{F}$ 是有限的，所以最终该序列肯定会重复，显然最早出现重复的元素是 1。即存在整数 p，使得元素 1，2，3，…，p 各不相同，但 $p+1=1$，则称该有限域的特征值为 p。有限域的特征值必须是素数，其原因是：如果 p 可以分解为 a 和 b 的积，则存在表达式

$$\underbrace{(1+1+\cdots+1)}_{a\text{次}}\underbrace{(1+1+\cdots+1)}_{b\text{次}} = \underbrace{(1+1+\cdots+1)}_{ab\text{次}}$$

这个表达式意味着存在两个乘积为零的非零元素，这与域的定义相矛盾。任意两个大小相同的有限域都是同构的，也就是说，如果不考虑对域中元素的不同的命名方式，那么给定域的大小时，只存在一个相应的有限域。

$\mathbb{F}$ 上关于变元 x 的多项式可以表示为 $a(x) = \sum_{i \geq 0} a_i x^i$，系数 $a_i \in \mathbb{F}$，非零系数的个数有限。非零多项式的次数指的是满足 $a_d \neq 0$ 的最大整数 d。$\mathbb{F}[x]$ 表示 $\mathbb{F}$ 上所有关于变元 x 的多项式的集合；该集合对于多项式加法和乘法运算可构成环。

多项式 $p(x) \in \mathbb{F}[x]$ 可以看做是 $\mathbb{F}$ 上的一个函数。但是需要注意的是，多项式不同并不一定就表示对应的函数不同；例如，$\mathbb{F}_2$ 上的 $p(x) = x + x^2$ 和 $p(x) = 0$ 这两个多项式对应相同的函数。如果 $\mathbb{F}$ 上存在满足 $p(a) = 0$ 的 a，那么就说 a 是 $p(x)$ 的一个根。次数为 d 的多项式最多有 d 个根。

如果存在次数不为 0 且满足 $p(x) = a(x)\,b(x)$ 的多项式 $a(x)$ 和 $b(x)$，那么就说 $p(x)$ 可约。反之如果不存在这样的因式分解，那么就说 $p(x)$ 不可约。假设 $\mathbb{F}$ 是有限的，m 是任意正整数，则 $F[x]$ 中总是存在次数为 m 的不可约多项式。

令$a(x)\in\mathbb{F}[x]$不可约且$a(x)$的次数$m>1$，则$\mathbb{F}[x]/\langle a(x)\rangle$可以看做是模$a(x)$的多项式运算构成的多项式环，该环可构成大小为$|\mathbb{F}|^m$的域。需要注意的是，任意次数的不可约多项式都是存在的。以有限域$\mathbb{F}_p$为例，对于任意正整数m，我们都可以构建一个大小为p^m的域。实际上，可以使用以下方法构造该域：假设$\mathbb{F}_q$是一个由q个元素组成的有限域，则$q=p^m$，$\mathbb{F}_q$可以构造为$\mathbb{F}_p[x]/\langle a(x)\rangle$，其中$p$为素数，$m$为整数，$a(x)$为不可约多项式。

下面我们举例说明。容易证明，$a(x)=x^2+x+1$在$\mathbb{F}_2[x]$上不可约。在模$a(x)$的多项式运算下，$\mathbb{F}_2[x]$中的任意元素$p(x)$都可以归入4个剩余类$[0]$、$[1]$、$[x]$、$[1+x]$中的一个，也就是说，$p(x)$除以$a(x)$的余一定是0、1、x或$1+x$中的一个。模(x^2+x+1)后的加法运算表和乘法运算表如下所示：

+	[0]	[1]	[x]	[1+x]
[0]	[0]	[1]	[x]	[1+x]
[1]	[1]	[0]	[1+x]	[x]
[x]	[x]	[1+x]	[0]	[1]
[1+x]	[1+x]	[x]	[1]	[0]

×	[0]	[1]	[x]	[1+x]
[0]	[0]	[0]	[0]	[0]
[1]	[0]	[1]	[x]	[1+x]
[x]	[0]	[x]	[1+x]	[1]
[1+x]	[0]	[1+x]	[1]	[x]

例如$x^2 mod (x^2+x+1)=x+1$，因此有$[x][x]=[1+x]$。这就构成了有限域F_4，其他更大的域可以通过类似的方法求得。

有许多计算方法对于所有的域$\mathbb{F}$都是可行的，例如用于求解线性方程组的高斯－约旦（Gauss－Jordan）消去法。此外，域上的矩阵及行列式也遵循通常的运算规则，例如当方阵中的元素取自域$\mathbb{F}$时，当且仅当其行列式不为零时，方阵可逆。

1.10.2　多项式的零点和非零点

对于$\mathbb{F}[x_1, \cdots, x_n]$上的多元多项式来说，“次数”这个概念不像对于单元多项式的情况那么明确。其中的某一项$x_1^{d_1}x_2^{d_2}\cdots x_n^{d_n}$可以看做是次数为$d_1$的关于$x_1$的多项式，或是次数为$d_2$的关于$x_2$的多项式，依此类推，也可以认为其总次数等于$d_1+d_2+\cdots+d_n$，其中$d_i$是非负整数。同样地，多项式$p(x_1, \cdots, x_n)\in\mathbb{F}[x_1, \cdots, x_n]$关于$x_i$的次数指的是$p$中所有非零项的关于$x_i$的次数的最大值，$p$的总次数指的是$p$中所有非零项的

总次数的最大值。如果对于任意 i，p 的关于 x_i 的次数恒小于一个固定的整数 d，那么就说 p 的次数以 d 为上界。显然，p 的总次数以 nd 为上界。

下面将要给出的一个著名的引理（参见参考文献［31］中的引理2）与稀疏零点引理是等价的。

引理 1-2　令 $f \in \mathbb{F}[x_1, \cdots, x_n]$，其次数以 d 为上界，并令 $A \subseteq \mathbb{F}$ 表示$\mathbb{F}$上的 $d+1$ 个不同元素的集合。如果对于 A^n 中的所有 n 元组 $(a_1, \cdots, a_n)$ 都有 $f(a_1, \cdots, a_n)=0$，则 $f \equiv 0$，即 f 恒等于零多项式。

证明：我们对变量的个数 n 使用归纳法。$\mathbb{F}$ 上的最高次数为 d 的单元非零多项式最多有 d 个零点，从而可证明当 $n=1$ 时，引理 1-2 成立。

假设对于 $n \geqslant 1$，引理 1-2 对于$\mathbb{F}[x_1, \cdots, x_n]$ 上的所有次数以 d 为上界的多项式成立。令 $f \in \mathbb{F}[x_1, \cdots, x_{n+1}]$ 且 f 的次数以 d 为上界，则 f 可以表示为

$$f(x_1, \cdots, x_{n+1}) = \sum_{i=0}^{d} f_i(x_1, \cdots, x_n) x_{n+1}^i$$

其中，$f_i \in \mathbb{F}[x_1, \cdots, x_{n+1}]$ 且 f_i 的次数也以 d 为上界。如果对于任意 $(a_1, \cdots, a_{n+1}) \in A^{n+1}$，$f(a_1, \cdots, a_{n+1})=0$ 都成立，则关于 x_{n+1} 的多项式 $f(a_1, \cdots, a_n, x_{n+1})$（可通过取定一组 $(a_1, \cdots, a_n)$ 得到）最少有 $d+1$ 个零点，从而 $f(a_1, \cdots, a_n, x_{n+1})$ 恒等于零多项式。也就是说，对于任意 $(a_1, \cdots, a_n) \in A^n$，$f_i(a_1, \cdots, a_n)=0$ 恒成立。在该假设下，所有 f_i 都恒等于零多项式，也就是说 f 恒等于零多项式。

根据归纳法，引理 1-2 得证。

1.3.3 节中的稀疏零点引理是引理 1-2 在 $A=\mathbb{F}$ 时的反命题。

接下来给出的定理在对随机线性网络编码的分析中也起着重要作用。

定理 1-2（**Schwartz - Zipple** 定理）　令 $f \in \mathbb{F}[x_1, \cdots, x_n]$ 是一个总次数为 d 的非零多项式，如果从集合 $A \subseteq \mathbb{F}$ 中随机选取 n 个值 $r_1, \cdots, r_n$，则 $f(r_1, \cdots, r_n)=0$ 的概率最大为 $d/|A|$。

证明：（对 n 使用归纳法）当 $n=1$ 时，f 是最多有 d 个根值的单元多项式。对于集合 A 恰好包含全部 d 个根值的极限情况，随机选取的 r_1 是其中一个根值的概率为 $d/|A|$。

假设对于 $n-1$ 元多项式定理 1-2 成立，其中 $n>1$。将 $f(x_1, \cdots, x_n)$ 表示为关于 x_1 的多项式，即

$$f(x_1, \cdots, x_n) = \sum_{i=0}^{k} x_1^i f_i(x_2, \cdots, x_n)$$

其中，$k \leqslant d$ 是 f 关于 x_1 的次数，显然多项式 $f_k(x_2, \cdots, x_n)$ 不恒为零。f_k 的总次数不大于 $d-k$，因此 $f_k(r_2, \cdots, r_n)=0$ 的概率最大为 $(d-k)/|A|$。

当 $f_k(r_2, \cdots, r_n)$ 不等于0时，令

$$g(x_1) = f(x_1, r_2, \cdots, r_n) = \sum_{i=0}^{k} x_1^i f_i(r_2, \cdots, r_n)$$

该多项式的次数为 k 且不恒为零，因此 $g(r_1)=0$ 的概率最大为 $k/|A|$。

用B表示事件$f(r_1, \cdots, r_n)=0$，用C表示事件$f_k(r_2, \cdots, r_n)=0$，则$P[C] \leqslant (d-k)/|A|$且$P[B|\overline{C}] \leqslant k/|A|$，因此有

$$\begin{aligned}P[B] &= P[B|C]P[C] + P[B|\overline{C}]P[\overline{C}] \\ &\leqslant P[C] + P[B|\overline{C}] \\ &\leqslant (d-k)/|A| + k/|A| \\ &= d/|A|\end{aligned}$$

Ho 等人在参考文献［16］中对 Schwartz－Zippel 定理作出了如下改进。

定理 1-3　令$f \in \mathbb{F}[x_1, \cdots, x_n]$是一个不恒为零的多项式，其次数上界为$d$，总次数上界为$d\eta$。如果从集合$A \subseteq \mathbb{F}$中随机选取$n$个值$r_1, \cdots, r_n$，$|A| \geqslant d$，则$f(r_1, \cdots, r_n)=0$的概率最大为$1-(1-d/|A|)^\eta$。

1.10.3　det（G_t）的次数

我们先介绍以下引理。

引理 1-3　令$\boldsymbol{A}$和$\boldsymbol{B}$是定义在某域上的$n \times n$阶矩阵，x是变元，则$\det(\boldsymbol{A}_x+\boldsymbol{B})$是关于$x$的次数不大于 rank（$\boldsymbol{A}$）的多项式。

证明：当$\boldsymbol{A}$的秩为0时，命题显然成立。

当$\boldsymbol{A}$的秩为1时，令$\boldsymbol{A}=[\boldsymbol{a}_1, \cdots, \boldsymbol{a}_n]$，$\boldsymbol{B}=[\boldsymbol{b}_1, \cdots, \boldsymbol{b}_n]$，其中$\boldsymbol{a}_i$和$\boldsymbol{b}_i$分别是$\boldsymbol{A}$和$\boldsymbol{B}$的第$i$列。考虑到矩阵的行列式对于其各行和各列具有多重线性，因此有

$$\begin{aligned}\det(\boldsymbol{A}_x+\boldsymbol{B}) &= \det(\boldsymbol{a}_1 x+\boldsymbol{b}_1, \cdots, \boldsymbol{a}_n x+\boldsymbol{b}_n) \\ &= \det(\boldsymbol{b}_1, \cdots, \boldsymbol{b}_n) + x\det(\boldsymbol{a}_1, \boldsymbol{b}_2, \cdots, \boldsymbol{b}_n) \\ &\quad + x\det(\boldsymbol{b}_1, \boldsymbol{a}_2, \boldsymbol{b}_3, \cdots, \boldsymbol{b}_n) + \cdots \\ &\quad + x\det(\boldsymbol{b}_1, \cdots, \boldsymbol{b}_{n-1}, \boldsymbol{a}_n) + \cdots\end{aligned}$$

又因为 rank（$\boldsymbol{A}$）＝1，因此该行列式展开式中包含的矩阵$\boldsymbol{A}$的列的数量不大于1，故该行列式关于x的次数不大于1。

当$\boldsymbol{A}$的秩$r>1$时，使用类似的方法可以证明$\det(\boldsymbol{A}_x+\boldsymbol{B})$的多重线性展开式中包含的矩阵$\boldsymbol{A}$的列的数量不大于$r$，即该行列式关于$x$的次数不大于$r$。

定理 1-4　假设$\boldsymbol{G}_t(\alpha_1, \cdots, \alpha_\eta)$是如 1.3.2 节中定义的无环网络中的全局转移矩阵，则多项式$\det(\boldsymbol{G}_t)$的最高次数为1。

证明：$\boldsymbol{G}_t$的第（i，j）项表示G'中从入节点i到出节点j（位于目的节点t上）的所有路径中的每条路径的分支增益之积的和。令e表示线图G'中分支增益为α_i的那条边。按照是否包含e，可以将这些路径划分为两个不相交的集合，则$\boldsymbol{G}_t$可以表示为

$$\boldsymbol{G}_t = \alpha_i \boldsymbol{x}\boldsymbol{y}^{\mathrm{T}} + \boldsymbol{H}$$

其中，$r \times r$阶矩阵$\boldsymbol{H}$可以通过将α_i置为0求得；$r \times 1$维向量$\boldsymbol{x}$表示从入节点到e的增益；$r \times 1$维向量$\boldsymbol{y}$表示从e到出节点的增益。注意到$\boldsymbol{H}$关于α_i是恒定的，α_i的作用是将矩阵的秩变成1。根据引理 1-3（令$x=\alpha_i$，$\boldsymbol{A}=\boldsymbol{x}\boldsymbol{y}^{\mathrm{T}}$以及$\boldsymbol{B}=\boldsymbol{H}$），本定理得证。

参考文献

[1] R. Ahlswede, N. Cai, S. -Y. R. Li, and R. W. Yeung, "Network information flow," IEEE Trans. on Inform. Theory, vol. 46, pp. 1204-1216, July 2000.

[2] R. W. Yeung, S. -Y. R. Li, N. Cai, and Z. Zhang, "Network coding theory," Foundations and Trends in Communications and Information Theory, vol. 2, no. 4, 5, pp. 241-381, 2006.

[3] C. Fragouli and E. Soljanin, "Network coding fundamentals," Foundations and Trens in Networking, vol. 2, no. 1, pp. 1-133, 2007.

[4] C. Fragouli and E. Soljanin, "Network coding applications," Foundations and Trends in Networking, vol. 2, no. 2, pp. 135-269, 2007.

[5] T. Ho and D. S. Lun, Network Coding: An Introduction. Cambridge, UK: Cambridge University Press, 2008.

[6] T. M. Cover and J. A > Tomas, Elements of Information Theory, Hoboken, NJ: John Wiley and Sons, second ed., 2006.

[7] K. Menger, "Zur allgemeinen Kurventheorie," Fundameta Mathematicae, vol. 10, pp. 96-115, 1927.

[8] L. R. Ford, Jr. and D. R. Fulkerson, "Maximal flow through a network," Canadian J. Mathematics, vol. 8, pp. 399-404, 1956.

[9] P. Elias, A. Feinstein, and C. E. Shannon, "A note on the maximum flow through a network," IRE Trans. on Inform. Theory, vol. 2, no. 4, pp. 117-119, 1956.

[10] S. -Y. R. Li, R. W. Yeung, and N. Cai, "Linear network coding," IEEE Trans. on Inform. Theory, vol. 49, pp. 371-381, Feb. 2003.

[11] R. Koetter and M. Medard, "An algebraic approach to network coding," IEEE/ACM Trans. on Networking, vol. 11, pp. 782-795, Oct. 2003.

[12] D. B. West, Introduction to Graph Theory. Upper Saddle River, NJ: Prentice Hall, second ed., 2001.

[13] S. Jaggi, P. Sanders, P. A. Chou, M. Effros, S. Egner, K. Jain, and L. M. G. M. Tolhuizen, "Polynomial time algorithms for multicast network code construction," IEE Trans. on Inform THEory, vol. 51, pp. 1973-1982, June 2005.

[14] N. J. A. Harvey, D. R. Karger, and K. Murota "Deterministic network coding by matrix completion," in Proc. 16th Annual ACM-SIAM Symp. on Discrete Algorithms (SODA), (Vancouver, BC), pp. 489-498, Jan. 23-25, 2005.

[15] A. I. Barbero and O. Ytrehus "Heuristic algorithms for small field multicast encoding," in Proc. 2006 IEEE Inform. Theory Workshop, (Chengdu, China), pp. 428-432, Oct. 22-26, 2006.

[16] T. Ho. M. Medard, R. Koetter, D. R. Karger, M. Effros, J. Shi, and B. Leong, "A random linear network coding approach to multicast," IEEE Trans. on Inform. Theory, vol. 52, pp. 4413-4430, Oct. 2006.

[17] R. M. Roth, Introduction to Coding Theory. Cambridge, UK: Cambridge University Press, 2006.

[18] K. Jain, M. Mahdian, and M. R. Salavatipour, "Packing Steiner trees," in Proc. 14th Annual ACM - SIAM Symp. on Discrete Algorithms (SODA), (Baltmore, MD), pp. 266 - 274, Jan. 12 - 14, 2003.

[19] A. Agarwal and M. Charikar, "On the advantage of network coding for improving network throughput," in Proc. 2004 IEEE Inform. Theory Workshop, (San Antonio, TX), pp. 247 - 249, Oct. 24 - 29, 2004.

[20] E. Halperin, G. Kortsarz, R. Krauthgamer, A. Srinivasan, and N. Wang, Integrality ratio for group Steiner trees and directed Steiner trees, "in Proc. 14th Annual ACM - SIAM Symp. on Discrete Algorithms (SODA), (Baltimore, MD), pp. 275 - 284, Jan. 12 - 14, 2003.

[21] L. Zosin and S. Khuller, "On dircted Steiner trees," in Proc. 13th Annual ACM - SIAM Symp. on Discrete Algorithms (SODA) 63, (San Francisco, CA), pp. 59 - 63, Jan. 6 - 8, 2002.

[22] C. Chekuri, C. Fragouli, and E. Soljanin, "On average throughput and alphabet size in network coding," IEEE Trans. on Inform, Theory, vol. 52, pp. 2410 - 2424, June 2006.

[23] P. A. Chou, Y. Wu, and K. Jain, "Practical network coding," in Proc. 2003 Allerton Conf. on Commun., Control, and Computing, (Monticello, IL), pp. 40 - 49, Oct. 1 - 3, 2003.

[24] R. Kotter and F. R. Kschischang, "Coding for errors and erasures in random network coding," IEEE Trans. on Inform. Theory, vol. 54, pp. 3579 - 3591, Aug. 2008.

[25] D. Silva, F. R. Kshischang, and R. Keotter, "A rank - metric approach to error control in random network coding," IEEE Trans. on Inform. Theory, vol. 54, pp. 3951 - 3967, Sept. 2008.

[26] E. M. Gabidulin, "Theory of codes with maxium rank distance," Probl. Peredachi Informatsii, vol. 21, no. 1, pp. 3 - 16, 1985.

[27] D. Silva and F. R. Kshischang, "Universal secure network coding via rank - metric codes," IEEE Trans. on Inform. Theory, vol. 57, no. 2, pp. 1124 - 1135, Feb. 2011.

[28] A. Rasala Lehman and E. Lehman, "Complexity classification of network information flow problems," in Proc. 15th Annual ACMSIAM Symp. on Discrete Algorithms (SODA), (New Orleans, LA), pp. 142 - 150, Jan. 11 - 14, 2004.

[29] S. Riis and R. Ahlswede, "Problems in network coding and error correcting codes," in Proc. First Workshop on Network Coding, Theory, and Applications, (Rival del Garda, Italy), Apr. 7, 2005.

[30] R. Dougherty, C. Freiling, and K. Zeger, "Insufficiency of linear coding in network information flow," IEEE Trans. on Inform. Theory, vol. 51, pp. 2745 - 2759, Aug. 2005.

[31] N. Alon, "Combinational nullstellensatz," Combinatorics, Probability and Computing, vol. 8, pp. 7 - 29, 1999.

第2章　无线系统中的网络编码

Dina Katabi　美国　马萨诸塞州　麻省理工学院　计算机科学与人工智能实验室
Sachin Katti　美国　加利福尼亚州　斯坦福大学　电子工程系
Hariharan Rahul　美国　马萨诸塞州　麻省理工学院　计算机科学与人工智能实验室

摘要

无线网络中的通信存在种种问题，如低吞吐量、盲区、干扰等。然而，无线网络也具有一些独特的特点，如无线媒介的广播特性、空间分集以及大量的数据冗余等，这些特点为设计解决上述问题的新理念提供了机会。本章将阐述网络编码在提高无线网络的吞吐量和可靠性方面带来的帮助。

关键词

无线网络、网状网络、实证、系统架构、网络协议。

2.1　引言

对无线网络的设计是以有线网络为蓝图的，首先将无线信道抽象为点到点链路，再将有线网络协议移植到无线环境中。例如，使用最短路径协议路由，路由器只对数据包进行转发但不修改数据，网络的可靠性依赖于重传。对于有线网络这样的设计可以良好地工作，但是对于不可靠且不可预知的无线媒介，效果就不那么好了。

无线媒介与有线媒介具有本质上的区别。有线链路可靠且可预测，而无线链路的误比特率很高，并且其特性会在很短的时间间隔内变化；有线链路是单播链路，而大部分的（使用全向天线的）无线链路都是广播链路；有线网络中的传输彼此之间不会相互干扰，而对于无线媒介中的通信来说干扰是十分常见的；有线节点通常是静态的，而无线节点则被设计为支持移动性和便携性。对于有线网络的设计与无线媒介的特性是冲突的，这导致了目前在无线网络中通信需要忍受低吞吐量、盲区以及难以支持无线节点的移动性等问题。

无线网络的这些特点乍一看似乎都是不利的，但是有更新的观点表明，通过全新的设计，人们可以对其中的一些特点加以利用。例如，无线媒介的广播特性有助于提高网络的可靠性。当节点广播某个数据包时，它附近的节点中很可能有节点会收到这个数据包，那么接收到数据包的节点就可以作为源节点的下一跳对该数据包进行转发。与此形成鲜明对比的是，依照目前的无线设计，我们会指定一个下一跳节点，如果这个节点没有接收到这个数据包，则上一跳节点就需要重传。上述属性被称为空间分集。一个简单的利用空间分集的方法是让所有接收到某个数据包的节点都将这个数据包向目的节点的方向进行转发，但这可能会导致多个节点不必要地转发相同的数据包并浪费网络带宽。

理想情况下，我们希望中继节点可以在转发数据包之前对它们进行分布式编码，从而使得所有的发送信息都有效。

此外，无线网络中还存在着大量的数据冗余，也就是说有大量的节点可用信息是重复的。首先，当数据包需要经由多跳节点时，它的内容将被它经由的多个节点掌握；其次，无线媒介的广播作用使得每一跳节点都把相同的数据包传送给它的无线电射程内的所有节点，这将放大这些冗余。因此，节点在转发某个数据包时，可以利用它附近的多个节点已经掌握该数据包内容的这一事实，也就是利用这些节点的边信息。人们希望以利用边信息的方式对传输进行编码，从而实现网络内数据压缩。

那么是否可以利用无线网络的一些固有的特点，比如空间分集和数据冗余等来设计其中的网络编码，从而提高网络的吞吐量和可靠性？本章概述了最近的一些在利用网络编码提高无线网络的吞吐量和可靠性方面的进展；并且主要关注实验性的无线系统，这对于其他章节中讨论的理论性工作具有补充意义；在网络编码基本思想的基础上，提出了可用于实际实现以及将网络编码集成到现有网络协议栈中的新算法，并通过原型实现和测试台评估证明了这些算法的实用性。

本章的重点是研究上述无线系统中的组播间编码、组播内编码和模拟网络编码，并对获得的性能增益进行实证。

2.2　网络编码背景：业内视点

传统的计算机网络采用存储 - 转发设计，即路由器缓存接收到的数据包并不作任何修改地将其转发。网络编码通过一个简单而基本的方式改变了这种设计。不论数据包是否来自相关的数据流，网络编码都允许路由器在转发数据包之前将其编码或混合。理论分析和实际测试都表明，网络编码可以显著地提高无线网络的吞吐量和可靠性[5, 14, 17, 21]。

我们可以将网络编码分为三类：组播内编码、组播间编码以及模拟网络编码。

(1) 组播内网络编码（Intra - session Network Coding）

路由器只混合去往相同目的节点的数据包且无需对编码包解码，目的节点在接收到足够多的编码包后进行解码操作[2]。关于网络编码的早期工作主要是对组播内网络编码的理论研究，有两个理论成果具有重要的现实意义。其一，结合参考文献［12，18，19］可知，对于多播通信，使用线性码可以实现其最大容量，且编码和解码可以在多项式时间内完成（这里的线性码指的是路由器产生的编码包，是其接收到的数据包的线性组合 $\sum c_i p_i$，其中 p_i 表示数据包，c_i 表示编码系数）。其二，Ho 等人证明了当随机选取编码系数时上述结论仍然成立[10]，这使得不同路由器之间无需协商对码字的选择的分布式网络编码成为可能。

组播内网络编码已经被广泛应用到了包括能量[8, 28]、保密[4, 11]、内容分发[6]和分布式存储[13]在内的诸多领域，本章主要关注它在无线网络中的应用[5, 21, 26]。我们将重点

讨论两个协议：MORE[5]和MIXIT[15]。前者是最早的组播内网络编码在无线网络中的实现，后者是一个关于组播内网络编码的跨层方案，该方案允许网络中的节点充分利用所有正确接收到的字节，甚至是被损坏的数据包里的正确的字节。

（2）组播间网络编码（Inter - session Network Coding）

路由器对来自交汇于网络的某个中间节点的不同的通信流的数据包进行混合或编码。如图2-1中给出的示例，Alice和Bob想要交换一对数据包，由于其无线覆盖范围不允许他们直接通信，因此需要通过路由器传输。使用路由方案时，Alice把她的数据包发送给路由器，再由路由器转发给Bob，接着Bob也把他的数据包发送给路由器，再由路由器转发给Alice。然而正如参考文献［16，17］中提到的，使用网络编码方案时，Alice和Bob分别把他们的数据包发送给路由器，路由器可以对这两个数据包求XOR（异或）并广播得到的编码包。Alice通过对编码包和她自己的数据包再次求XOR可以恢复出Bob的数据包，Bob也可通过同样的方法恢复出Alice的数据包。上述过程利用网络中的数据冗余对信息进行压缩，在一次发送中提供两个数据包从而提高吞吐量。上述思想在参考文献［27］中首次被提出，进而在各种网络拓扑和通信模式下被推广[17]，并产生了网络编码的首次实现——COPE协议，该协议证明了网络编码在实际应用中带来的显著的吞吐量增益。

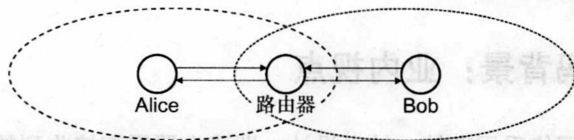


图2-1　关于网络编码可以提高吞吐量的一个简单示例。使用网络编码时，Alice和Bob只需经过3次传输就可以交换一对数据包

与组播内网络编码相比较，组播间网络编码是作为一个实际的协议与系统体系结构被提出的[17]。它的实践成功激发了大量的旨在描述其吞吐量界限的理论工作[20, 25]。

（3）模拟网络编码（ANC）

模拟网络编码（ANC）将网络编码操作从数据包的层面延伸到了模拟信号的层面[14]。如果两个数据包被同时发送，则接收端接收到的被干扰的信号将是两个原始信号的线性组合。模拟网络编码不是通过路由器对数据包进行混合或编码，而是直接借助信道对数据包的内容进行混合。模拟网络编码允许多个节点同时发送信号，因此可以减少发送次数。

模拟网络编码也被称为物理层网络编码，它在参考文献［24］中首次被提出。然而参考文献［24］中对模拟网络编码的设计是基于符号级同步、载波 - 频率同步以及载波 - 相位同步的假设，这些在实际中都是不可能实现的。在参考文献［14］中，Katti等人假设发射机之间不同步并在不考虑信道失真的条件下提出了相应的解码算法。在该算法的基础上，他们首次实际实现了模拟网络编码。

值得注意的是，对于组播内以及组播间的情况，均可以使用模拟网络编码，相关内容将在本章的后续部分讨论。

表 2-1 中对组播内网络编码和组播间网络编码进行了比较。

表 2-1　不同网络编码方法的比较

组播内网络编码	组播间网络编码	模拟网络编码
对同一组播流的数据包编码	对不同组播流的数据包编码	通过同时发送多个信号实现编码
在目的节点对数据包解码	路由器通常需要对数据包进行解码和再编码	路由器无需解码，而是放大混合信号并将其转发
常用于多播应用	常用于单播应用	用于单播及多播应用
适用于链路有损、拓扑结构不可预测的网络	适用于低损、拓扑结构稳定的网络	适用于 SNR 较高的信道
提高可靠性	提高吞吐量	提高吞吐量

2.3　网络编码在无线网络中的应用

以下各节描述了围绕网络编码建立未来无线网络的各种潜在的好处。我们将从无线网络的不同角度展开讨论，例如吞吐量、可靠性以及对干扰的处理，并给出不同的网络编码方法。

2.3.1　COPE：提高吞吐量的网络编码

现有的无线网络在吞吐量方面还有很大的提升空间。本节简要介绍了 COPE——以可提高无线吞吐量的网络编码为中心设计的一个无线架构。在不对网络的拓扑结构、传输需求及调度方式作任何假设的条件下，COPE 是最早提出的可对来自多个链接的数据包进行编码的通用算法。它不仅首次将网络编码集成到现有的网络协议栈设计中，而且还首次在测试平台上实现并评估了网络编码[16, 17]。

COPE 的基本思想是：当路由器已知网络中的各个节点对信息的接收情况时，它可以通过网络编码对待发送的信息进行压缩，从而提高无线吞吐量，例如图 2-1 中的 Alice - Bob 场景。通过对每个邻居节点已知的信息和其他邻居节点需要的信息进行匹配，路由器可以在一次发送中同时给多个不同的邻居节点提供不同的数据包。由于被编码的数据包的下一跳节点各不相同，即这些数据包来自不同的组播流，因此称这种编码方式为组播间网络编码。

COPE 不仅限于图 2-1 中的两个节点在反向路径上的通信，还可以做进一步的延伸。它利用了无线媒介的共享特性，即无线媒介可以无代价地将数据包广播给与其路径相邻的整个区域。由于每个区域中的节点用于解码的数据包集合之间有很大的重复部分，因此这种特性为编码创造了一个有利的环境。例如，图 2-2 中说明了如何把 2 个经由不同路径的连接编码到一起。假设源节点 S_1 和 S_2 分别想要发送数据包流给目的节点 D_1 和 D_2。由于无线媒介是广播媒介，因此节点 D_1 能够收听到 S_2 发出的数据包，而 D_2

能够收听到 S_1 发出的数据包。利用这一点，路由器 R 可以对 S_1 和 S_2 的数据包进行 XOR 运算，并广播得到的编码包，该编码包对于两个目的节点都是有效的。图 2-2 中的“X 型”拓扑结构是 Alice - Bob 拓扑结构的扩展。这是一个重要的扩展，因为在实际的无线网络中只有极少数的流会经由像 Alice - Bob 那样的反向路径，更多的流会在某个路由器上交汇从而被一起编码。

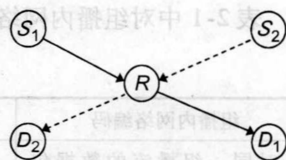


图 2-2　“X 型”拓扑：两个数据包流在 R 点交汇

对于更一般的拓扑结构，COPE 可以比上例中节省更多的带宽。它能够对多个数据包进行 XOR 运算，从而成倍地增长吞吐量。

2.3.1.1　COPE 协议

COPE 给无线网状网络提供了一个新的转发架构，它在 IP 层和 MAC 层之间插入了一个负责检测并利用编码机会的编码层，目的是在单次发送中实现多个数据包的传递。COPE 协议主要采用了以下 3 种技术。

（1）机会收听

COPE 要求节点收听并存储那些可能会对解码有用的数据包，这些数据包有两类来源：

- 节点自身的数据包：例如图 2-1 所示的 Alice - Bob 场景，Alice 和 Bob 都保留了它们发送给路由器的数据包的副本，他们可以用这些副本对来自路由器的编码包进行解码。
- 节点监听到的数据包：无线媒介的广播特性给节点创造了许多监听数据包的机会。依照目前的传输协议，节点会丢弃它监听到的目的节点不是自己的数据包。而在 COPE 协议下，节点始终处于一种混杂模式，它们监听无线媒介中发生的所有通信并把所有监听到的数据包都存储一段有限的时间 T，T 的值应大于网络的最大单向时延（默认值为 $T=0.5\text{s}$）。

（2）机会编码

节点编码的目标是最大化一次发送过程中传递的数据包的数量，同时确保所有的下一跳节点都有足够的信息解码其所需的数据包，因此关键问题在于把哪些数据包编码到一起能够最大化吞吐量。

通过一个例子可以很好地说明这个问题。如图 2-3a 所示，节点 B 的输出队列中有 4 个数据包 p_1、p_2、p_3 和 p_4，它的邻居节点已经各自监听到了这些数据包中的一部分。图 2-3b 中的列表给出了 B 的队列中各个数据包要去往的下一跳节点。使用路由方法时，B 会在被允许发送时从队列的最前端取出数据包 p_1 发送。假设 B 知道所有的邻居节点都有哪些数据包，B 可以使用的几种编码策略如图 2-3c 所示。

1）B 发送 $p_1 \oplus p_2$。由于 C 已经有 p_1，因此它可以通过对 p_1 和 $p_1 \oplus p_2$ 进行 XOR 运算获得它所需要的原始包 p_2。但是 A 没有 p_2 的信息，因此它不能对 $p_1 \oplus p_2$ 解码。也就是说，B 发送 $p_1 \oplus p_2$ 时只有一个邻居节点可以获得有效信息，因此这个编码策略对于 B 来说不是很好。

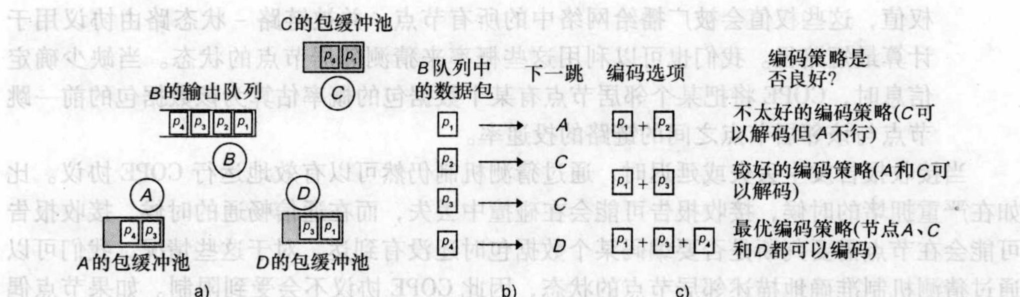


图 2-3　COPE 编码举例：节点 B 的发送队列中有 4 个包，图 b 中列出了这些包的下一跳节点。B 的每个邻居节点都在本地存有这些包中的一部分，如图 a 所示。节点 B 有多种编码策略见图 c，但它应该选择的是能够最大化在一次发送中传递的数据包数据量的第 3 种编码策略

a）*B* 可以对它想要发送的数据包编码　b）各个数据包的下一跳节点　c）几种可能的编码选择

2）*B* 发送 $p_1 \oplus p_3$。此时邻居节点 *C* 和 *A* 可以从这一次传输中分别获得它们所需的数据包，因此这个编码策略比第 1 种编码策略要稍好一些。

3）*B* 发送 $p_1 \oplus p_3 \oplus p_4$。此时 3 个邻居节点都可以从这一次传输中分别获得它们所需的数据包，例如，*A* 对接收到的编码包和存储在本地的 p_3、p_4 进行 XOR 运算可以恢复出它所需要的 p_1，相似地，*C* 和 *D* 也可以分别恢复出它们所需要的数据包，因此这是 *B* 的最佳编码策略。

上述例子强调了一个重要的编码问题：在某个中间节点上，可以把来自不同单播流的数据包编码到一起。由于这些数据包的下一跳节点不相同，因此它们需要能够在下一跳被解码，否则数据包将被转发给不需要它们的节点，这将浪费宝贵的容量。编码算法必须确保所有编码包的下一跳节点都可以解码得到它们所需要的数据包，可以通过下述规则实现这一点：

如果节点要发送 n 个数据包 p_1，…，p_n 给 n 个下一跳节点 r_1，…，r_n，那么只有当每个下一跳节点 r_i 都已经有 $n-1$ 个数据包 p_j（$j \neq i$）时，节点才能对这 n 个数据包一起进行 XOR 运算。

上述规则确保了所有下一跳节点都能对编码包解码并获得它们需要的数据包。为了使编码效益最大化，节点在有机会发送数据包时应优先选择满足上述规则的最大的 n。

（3）掌握邻居节点状态

要实现上述编码算法，COPE 要求节点掌握它们的邻居节点的本地数据包信息。为了实现这一点，COPE 采用了以下两个机制：

- 接收报告：通过广播接收报告，节点可以告知邻居节点本地存储了哪些数据包。接收报告以注释的形式和数据包一起发送。当节点没有数据包需要发送时，它会以特殊控制包的形式周期性地发送接收报告。
- 猜测：如果节点没有收到其转发队列中的某个数据包的接收报告，它将猜测它的邻居节点是否有这个数据包。COPE 利用路由计算来实现更加智能的猜测。无线路由协议能够计算每对节点之间的投递率并通过它来识别优质链路，例如使用 ETX 矩阵[7]可以周期性地计算投递率并给每条链路分配一个等于 1/投递率的

权值，这些权值会被广播给网络中的所有节点，并被链路-状态路由协议用于计算最短路径。我们也可以利用这些概率来猜测邻居节点的状态。当缺少确定信息时，COPE 将把某个邻居节点有某个数据包的概率估算为该数据包的前一跳节点与该邻居节点之间的链路的投递率。

当接收报告发生丢失或延迟时，通过猜测机制仍然可以有效地运行 COPE 协议。比如在严重拥堵的时候，接收报告可能会在碰撞中丢失，而在通信畅通的时候，接收报告可能会在节点需要判断是否要编码某个数据包时还没有到达。对于这些情况，我们可以通过猜测机制准确地描述邻居节点的状态，因此 COPE 协议不会受到限制。如果节点偶然地猜测错误导致编码包在某些下一跳节点不能被解码，则相应的原始包会被重新发送并有可能与新的一组原始包一起被编码。

2.3.1.2　性能结果

理论上 COPE 协议可以减少一半的传输[17]，即翻倍增长吞吐量。然而 Katti 等人在参考文献［17］中的研究显示，在实际环境中 COPE 能够获得更大的吞吐量增益。对于由 20 个节点组成的网状网络测试平台，其总吞吐量作为通信需求的函数如图 2-4 所示。从图中我们看到，相比于目前的 802.11 协议，COPE 协议带来了 3 ~4 倍的吞吐量提升。实际吞吐量增益比理论增益要大的原因是，COPE 协议可以缓解某些网络节点过于繁忙的情况。例如 Ad Hoc 网络中的大部分路径都会在网络的中心处交汇，因此位于网络中心的节点可能会经历拥塞，它们会丢弃溢出队列的数据包。这些被丢弃的数据包在到达中心节点的过程中已经消耗了网络的带宽资源，在中心节点上丢弃它们会浪费网络资源并导致整个网络的吞吐量显著降低。与此相反的是，使用网络编码时位于网络中心的拥塞节点可以在一次发送中传输多个数据包，这将加快它们对包队列的处理，从而减少对数据包的丢弃。

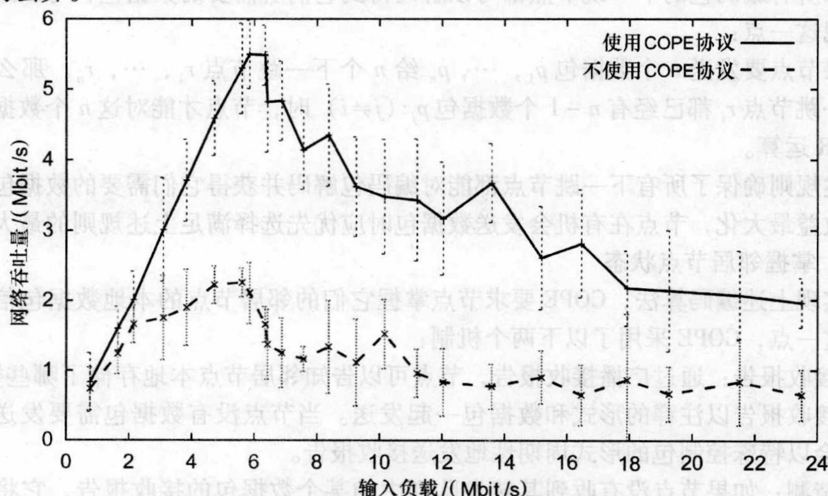


图 2-4　对于无线 Ad Hoc 网络，使用组播间网络编码协议 COPE 可以获得若干倍的吞吐量增益，图中所示的是对一组 UDP 数据包流的仿真结果，假设我们随机选取源-目的节点组，数据包的到达时间服从泊松分布，数据包的大小服从重尾分布

2.3.2　MORE：提高可靠性的网络编码

在目前的无线架构中保障可靠性的主要方法是重传丢失的数据包。这个方法适用于误比特率非常低的有线网络，但是对于容易出现传输错误的无线信道却是十分低效的。

本节将介绍组播内网络编码并说明它能够有效地保证传输可靠性的原理，本节与 2.3.1 节的区别在于路由器仅对前往相同目的节点的数据包进行混合。

我们将重点介绍 MORE 协议[5]，它是最早实现的组播内网络编码协议，也是一个机会路由协议。城市结构的错综复杂以及诸如局部 WLAN 网络[1]等的众多干扰源导致室外无线网状网络的信道质量往往很差。传统的路由协议在发送数据包前会先选择它的下一跳节点，然而当信道质量很差时，所选择的下一跳节点能够接收到这个数据包的概率很小。与此相反的是，机会路由协议允许所有节点监听传输并允许距离目的节点较近的节点参与数据包的转发[3]，从而能够显著地提高对丢包的容错性。然而机会路由机制也引入了一些问题：当多个节点都监听到广播的数据包时，可能会导致对同一个数据包的不必要的转发。这使得机会路由需要预先全局协商好哪个节点负责转发哪个数据包。不同的是，MORE 协议无需节点之间的协商，而是通过引入网络编码更具伸缩性地解决了这一问题。在说明 MORE 协议的工作机制之前我们先引入一些例子来阐明它的设计。

2.3.2.1　示例 1：盲区

网络编码有助于处理盲区问题。考虑图 2-5a 中的场景，Alice 想要发送一个文件给 Bob，Bob 不在 Alice 的直接通信范围内，因此 Alice 需要借助于中间节点来完成通信。附近有 5 个无线节点可以将 Alice 的数据包转发给 Bob。不幸的是，Alice 位于盲区内，她附近的所有无线节点与她通信时的丢包率都是 80%。使用 802.11 协议时，Alice 将选择一条最佳路径来完成她到 Bob 的传输。但由于所有的路径都是有损的，因此每个数据包都需要发送 6 次（从 Alice 到中继节点需要发送 5 次，从中继节点到 Bob 需要发送 1 次）。

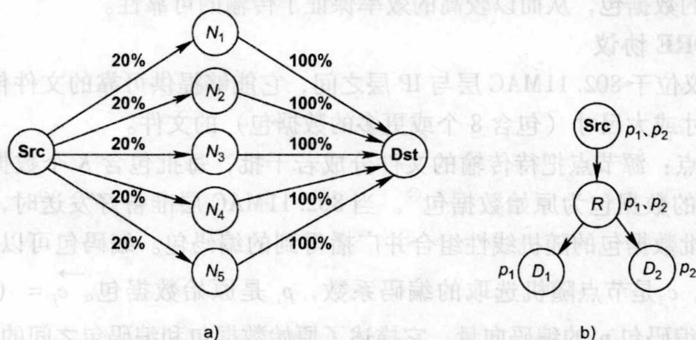


图 2-5　关于网络编码能够提供可靠性增益的举例

a）网络编码允许所有的路由器参与转发并且还不会导致重复传输

b）通过网络编码，多播网络中的接收节点可以有效地恢复出丢失的数据包

一个更好的方法是利用空间分集来提高 Alice 传输的吞吐量。Alice 广播要发送的数据包，所有监听到的节点都可以把数据包转发给 Bob。这种方法将一个数据包被成功发送给中继节点的概率从 20% 提高到了 $(1-0.8^5)\times 100\approx 67.2\%$。因此，平均每个数据

包只需要发送 2.5 次（从 Alice 到中继节点需要发送 1.5 次，从中继节点到 Bob 需要发送 1 次），从而获得了 2.4 倍的吞吐量提升。然而只分集不编码会导致另一个问题：可能会有多个中继节点监听到同一个数据包并都尝试将该数据包转发给 Bob，这将导致杂散发射并浪费无线带宽。

把空间分集和网络编码组合到一起可以解决上述问题。Alice 广播数据包，所有中继节点都可以参与对 Alice 到 Bob 的数据包的转发。不同的是，中继节点需要构建关于它已经接收到的所有来自 Alice 的数据包的随机线性组合。具体地说，中继节点发送给 Bob 的数据包的形式是 $p'_i = \sum c_{ij}p_j$，其中 p_j 是某个来自 Alice 的数据包，c_{ij} 是随机编码系数。如果这个文件由 n 个数据包组成，那么 Bob 在收到任意 n 个编码包后可以如下解码得到整个文件㊀：

$$\begin{bmatrix} p_1 \\ \vdots \\ p_n \end{bmatrix} = \begin{bmatrix} c_{11} & \cdots & c_{1n} \\ \vdots & & \vdots \\ c_{n1} & \cdots & c_{nn} \end{bmatrix}^{-1} \begin{bmatrix} p'_1 \\ \vdots \\ p'_n \end{bmatrix} \tag{2-1}$$

Bob 一解码出整个文件就立刻广播确认信息，以通知中继节点停止发送。

2.3.2.2　示例 2：多播

考虑图 2-5b 中的示例，源节点想要把一个视频流多播给目的节点 D_1 和 D_2。假设源节点发送数据包 p_1 和 p_2 给路由器 R，R 又把这 2 个数据包广播给目的节点。由于不同节点之间的无线接收情况具有较大的独立性[22, 23]，因此对于 R 广播的 2 个数据包，可能 D_1 只收到了 p_1 而 D_2 只收到了 p_2。在这种情况下，R 需要重传 2 个数据包才能使 D_1 和 D_2 分别接收到它们丢失的数据包。但是使用网络编码时，我们允许 R 对 p_1 和 p_2 进行 XOR 运算（即 $p_1 \oplus p_2$）并广播得到的编码包。通过一次发送两个目的节点都可以获得它们丢失的数据包，从而以较高的效率保证了传输的可靠性。

2.3.2.3　MORE 协议

MORE 协议位于 802.11MAC 层与 IP 层之间，它能够提供可靠的文件传输，尤其适合传输中等尺寸或大尺寸（包含 8 个或更多的数据包）的文件。

（1）源节点：源节点把待传输的文件分成若干批，每批包含 K 个数据包，我们称这 K 个未编码的数据包为原始数据包㊁。当 802.11MAC 层准备好发送时，源节点将创建关于当前一批数据包的随机线性组合并广播得到的编码包。编码包可以表示为 $p'_j = \sum_i c_{ji}p_i$，其中，c_{ji} 是节点随机选取的编码系数，p_i 是原始数据包。$\vec{c_j} = (c_{j1}, \cdots, c_{j1}, \cdots, c_{jK})$ 称为编码包 p'_j 的编码向量，它描述了原始数据包和编码包之间的关系。

源节点在发送时会给每个数据包添加一个报头，报头的内容是该数据包的编码向量（解码时需要使用）、批次的 ID、源节点和目的节点的 IP 地址以及可以转发该数据包的

㊀　随机系数的选取需保证生成的编码包 p'_i 之间以指数级的概率[10]相互独立。

㊁　译者注：原词为 native packet，直译为本地数据包，译者以为“原始数据包”更符合其意义。

节点列表。发送节点会列出离目的节点的距离比自己近的所有节点，并把它们按照与目的节点之间的距离进行排序（这个距离可以通过 ETX 矩阵[7] 计算得到）。

发送节点将持续发送当前批次的数据包，直到接收到目的节点对该批数据包的确认信息它才会开始处理下一批数据包。

（2）转发节点

转发节点需要监听其无线覆盖范围内的所有数据传输。当节点监听到一个数据包时，它会检查自己是否位于该数据包的转发节点列表中。如果是，则该节点将继续检查该数据包是否包含新的信息，我们称包含新信息的数据包为新生数据包。严格意义上讲，如果一个数据包与节点已经接收到的同批次的所有数据包之间都是线性独立的，那么这个数据包就是新生的。

如果节点位于某个数据包的转发节点列表中，那么这个数据包的到达将会触发该节点广播一个编码包，也就是节点会创建一个关于它已经接收到的所有同批次的编码包的随机线性组合。需要注意的是，编码包的线性组合仍然是相应的原始数据包的线性组合。具体地说，假设转发节点已经接收到了编码包 $p'_j = \sum_i c_{ji} p_i$（p_i 是原始数据包），则该节点通过对这些编码包进行线性组合可以创建更多的编码包 $p'' = \sum_j r_j p'_j$，其中，r_j 是随机系数。最后得到的编码包 p'' 可以用原始数据包表示为 $p'' = \sum_j r_j (\sum_i c_{ji} p_i) = \sum_i (\sum_j r_j c_{ji}) p_i$，因此它也可以看做是原始数据包的线性组合。

（3）目的节点

目的节点对每个接收到的数据包都会检查它是否是新生的，即检查该数据包是否与已经接收到的数据包线性独立。非新生数据包由于没有包含新的信息，因此会被目的节点丢弃。在接收到 K 个新生数据包后，目的节点会通过式（2-1）中的矩阵求逆来解码出整批原始数据包。每当目的节点解码完一批原始数据包，它都会发送一个确认信息告知源节点可以开始传输下一批数据包。由于 MORE 协议采用了 802.11 标准并兼容最短路径路由，因此可以使用最优路径路由来传输 ACK 信息。

2.3.2.4　实证结果

参考文献［5］在测试平台上比较了 MORE 协议和其他两个多播协议，其结果如图 2-6 所示。其中 Srcr 协议以源节点为根建立了一个多播树，并把去往目的节点的多播流分配给该树的各条边；ExOR 协议利用无线媒介的广播特性把一个数据包同时发送给多个节点。无线测试平台由 20 个节点组成，图 2-6 中画出了平均每个目的节点的吞吐量。在测试场景中，源节点要把一个文件多播给若干目的节点。随着目的节点数量的增大，平均每个节点的吞吐量减少，这与我们的预期吻合。然而有趣的是，从图 2-6 中可以看出，MORE 协议带来的吞吐量增益随着目的节点数量的增大而增大。组播内网络编码（这里指 MORE 协议）相比于 ExOR 协议获得了 35% ~200% 的吞吐量增益，相比于 Srcr 协议获得了 100% ~300% 的吞吐量增益。实际上，这些吞吐量增益是通过提高传输的可靠性而获得的，也就是减少了将数据包交付给目的节点所需的传输次数。实证表明，

对于单播通信 MORE 协议也能够获得显著的增益[5]。

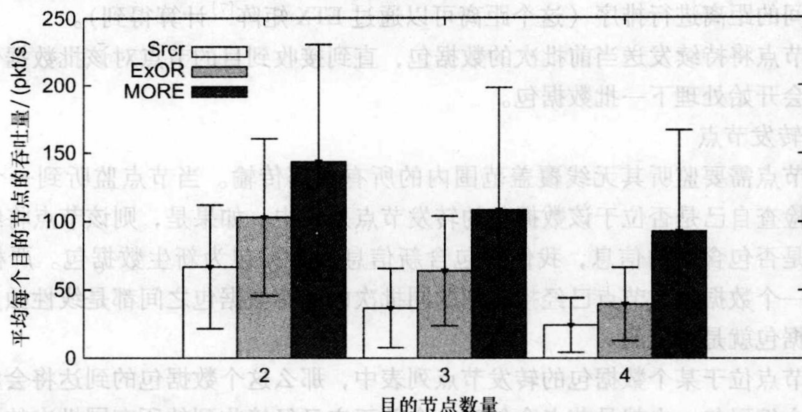


图 2-6　使用组播内网络编码协议——MORE 提高多播吞吐量。图中分别画出了使用 MORE. ExOR 和 Srcr 协议时平均每个目的节点的多播吞吐量。粗柱表示平均每个节点的吞吐量，细线表示标准差

2.3.3　模拟网络编码：利用无线干扰

在无线网络中通信时，为了防止相互干扰，我们会尽量避免同时进行多个传输。而模拟网络编码（ANC）的思路正好相反：它根据特定的策略挑选若干个发送者，并利用这些发送者之间的相互干扰来提高网络的吞吐量。当多个发送者同时发送时，数据包会产生碰撞。但是，当我们在信号层面进行更深入的研究时，会发现两个数据包的碰撞意味着它们的物理信号在信道中经过衰减和相移后发生相互叠加。因此，如果接收者知道干扰数据包的内容，它就可以在信道均衡后从叠加信号中减去干扰数据包信号，从而获得它想要的数据包信号，然后再用标准方法解调从而获得它想要的数据包。我们将通过以下示例详细地说明这个过程。

（1）交叠于路由器的信息流

再次考虑图 2-7 中的 Alice - Bob 场景。前面已经提到，采用组播间网络编码可以把传输时隙从 4 个减少到 3 个。节省下来的时隙可以用于发送新的数据，从而提高无线吞吐量。那么，能不能再进一步减少传输时隙呢？有没有可能在 2 个时隙内完成 2 个数据包的传输？

答案是肯定的。如果允许 Alice 和 Bob 的数据包在路由器上交叠，那么 Alice 和 Bob 就可以同时发送各自的数据包，一共占用 1 个时隙。由于信号发生了交叠，因此路由器接收到的是 Alice 和 Bob 的信号和 $s_A(t)+s_B(t)$（不考虑衰减和相移）。信号的交叠使得路由器不能把接收到的信号解调成比特形式。然而路由器并不需要进行解调，它可以只在物理层对接收到的交叠信号进行简单的放大和转发，这将占用第 2 个时隙。由于 Alice 知道她自己发送的数据包，也就知道相应的信号 $s_A(t)$，因此她可以从接收到的交叠信号中减去 $s_A(t)$ 以获得 $s_B(t)$，进而解调得到 Bob 发送的数据包。Bob 也可以用类似的

方法恢复出 Alice 发送的数据包。我们称上述方法为模拟网络编码。它与数字网络编码的原理相似，但它的操作对象是无线信道中的物理信号本身。模拟网络编码可以把传输 2 个数据包所需的时隙从 4 个减少到 2 个，从而使无线吞吐量得到翻倍的增长。

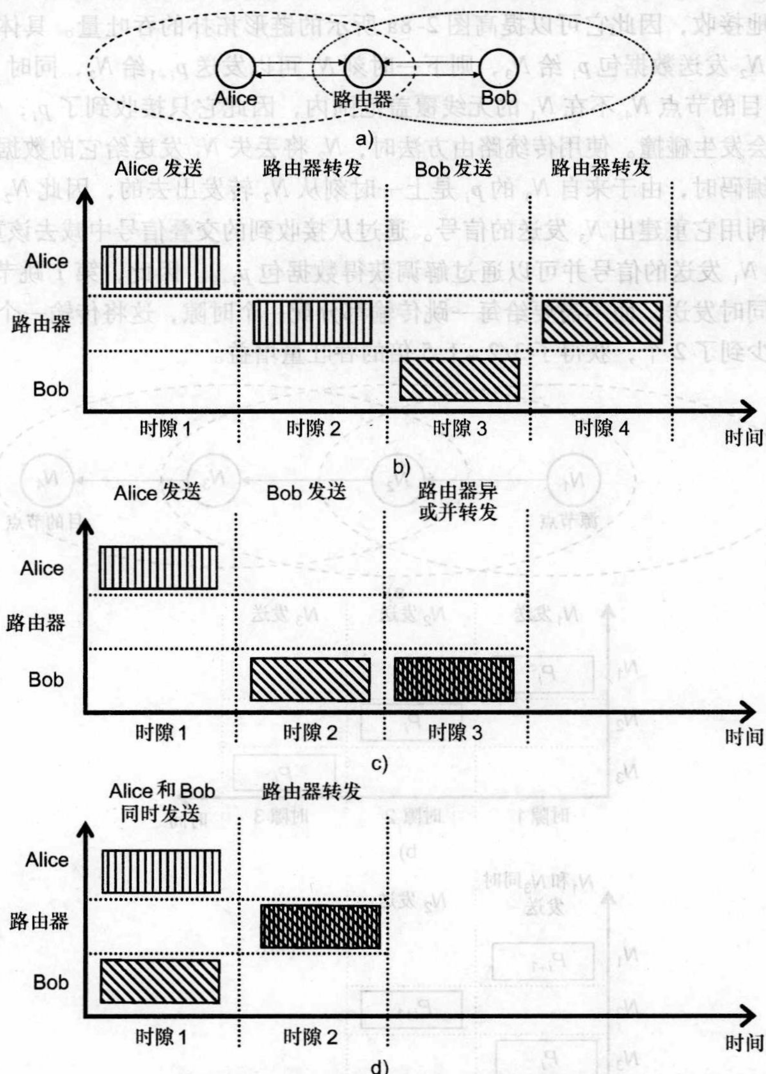


图 2-7　Alice – Bob 拓扑结构。使用模拟网络编码时，Alice 和 Bob 同时给路由器发送信号，两个信号流在路由器上交叠，路由器把交叠信号转发给 Alice 和 Bob，他们再分别解调出对方的数据包。模拟网络编码把传输时隙从 4 个减少到了 2 个，翻倍增长了吞吐量

a）Alice – Bob 拓扑结构　b）传统方法　c）数据网络编码　d）模拟网络编码

（2）单一方向的流

以上是关于组播间模拟网络编码的例子，模拟网络编码还能用于来自同一组播流的数据包。考虑图 2-8a 所示的链形拓扑结构，其中的数据流需要经过 3 跳节点。采用传

统的路由方法时，每个数据包从源节点被发送到目的节点都需要 3 个时隙。对于该情况，使用数字网络编码不能减少传输时隙，但使用模拟网络编码却可以。

模拟网络编码允许 N_1 和 N_3 同时发送，尽管 N_1 和 N_3 发送的信号会发生碰撞，但都可以被正确地接收，因此它可以提高图 2-8a 所示的链形拓扑的吞吐量。具体地说，假设某一时刻 N_2 发送数据包 p_i 给 N_3，则下一时刻 N_1 可以发送 p_{i+1} 给 N_2，同时 N_3 发送 p_i 给 N_4。由于目的节点 N_4 不在 N_1 的无线覆盖范围内，因此它只接收到了 p_i，但是在 N_2 上 p_i 和 p_{i+1} 会发生碰撞。使用传统路由方法时，N_2 将丢失 N_1 发送给它的数据包。而使用模拟网络编码时，由于来自 N_3 的 p_i 是上一时刻从 N_2 转发出去的，因此 N_2 已知该数据包并可以利用它重建出 N_3 发送的信号。通过从接收到的交叠信号中减去该重建信号，N_2 可以获得 N_1 发送的信号并可以通过解调获得数据包 p_{i+1}。因此，第 1 跳节点和第 3 跳节点可以同时发送，而不需要给每一跳传输都分配一个时隙，这将传输一个数据的时隙从 3 个减少到了 2 个，获得了 3/2 = 1.5 倍的吞吐量增益。

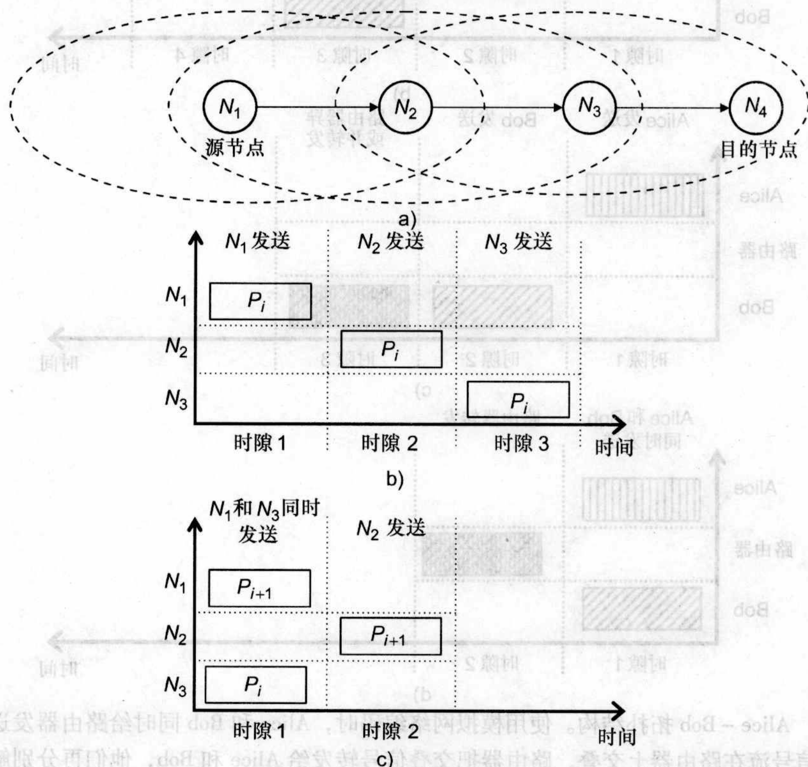


图 2-8 单向网络流的链形拓扑结构。N_1 和 N_3 可以同时发送，由于 N_2 已知交叠信号中 N_3 发送的那部分内容，因此 N_2 接收到交叠信号后可以恢复出 N_1 发送的数据包。传输一个数据包所需的时隙从 3 个减少到了 2 个，获得了 3/2 = 1.5 倍的吞吐量增益

a）链形拓扑结构 b）传统方法 c）模拟网络编码方法

需要注意的是，链形网络中使用的是组播内模拟网络编码，而 Alice - Bob 场景中展示的是一种组播间模拟网络编码。

ANC 解码器：下面介绍 ANC 解码器的基本原理，详细内容读者可以参阅参考文献［14］和［9］。再次考虑图 2-1 中的简单场景，Alice 接收到交叠信号后，要如何解出 Bob 的数据包呢？

首先我们要注意信号通过信道时会经历衰减和相移，而且两个发送机不可能精确地同时发送，因此这个交叠信号并不仅仅是将 Alice 的信号和 Bob 的信号简单相加得到，假设 Alice 接收到的信号是 $f_1(s_A(t)) + f_2(s_B(t-T))$，其中 $f_1()$ 和 $f_2()$ 表示信道函数，T 表示两个信号之间的时间偏移。要解码交叠信号，Alice 需要已知 $f_1()$，并在做减法前将这个函数作用于她的发送信号。那么在做减法后，Alice 剩下的就是 Bob 的信号 $f_2(s_B(t-T))$，这和仅传输一个 Bob 的信号的效果是相同的，因此 Alice 可以使用一些标准方法进行解码。

那么 Alice 要怎样获知她的信道函数呢？答案很简单：假设两个传输不完全同步，Alice 的信号中有一些比特先于 Bob 的信号发送，因此它们不会受到 Bob 的干扰。Alice 可以对这些比特使用标准方法来得到她的信道函数。

Bob 可以用相似的方法解码，唯一不同之处在于他的数据包传输要晚于 Alice，他的不受干扰的比特位于数据包的尾部。因此 Bob 可以接收所有的数字采样值，根据其中的无干扰采样值获得他的信道函数并进行信道补偿，最后从交叠信号中减去自己的信号以获得 Alice 的信号。

以上描述的是一个简化的解码器，读者可以参阅参考文献［9］中的 ZigZag 解码来了解更具一般性的解码器。无论采用什么调制方式，无论数据包有多大，ZigZag 都可以对相应的模拟网络编码进行解码。

2.4　结论

网络编码提高了无线网络的有效性、可延拓性以及可靠性。随之而来的是我们需要重新思考采用的 MAC 层、路由层以及传输层协议。近年来，在将网络编码集成到无线网络方面已经获得了显著的成果，并完成了许多能获得显著的吞吐量增益和可靠性增益的实际实现。我们相信未来的研究会继续沿着这些方向进行下去，并且会出现更多在无线网络系统中利用网络编码的新方法。

参考文献

[1] D. Aguayo, J. Bicket, S. Biswas, G. Judd, and R. Morris. Linklevel measurements from an 802. 11b mesh network. In ACM SIGCOMM, 2004.

[2] R. Ahlswede, N. Cai, S. R. Li and R. W. Yeung. Network information flow. In IEEE Transactions on Information Theory, July 2000.

[3] S. Biswas and R. Morris. Opportunistic routing in multi - hop wireless networks. ACM SIGCOMM, 2005.

[4] N. Cai and R. W. Yeung. Secure Network Coding. In ISIT, 2002.

[5] S. Chachulski, M. Jennings, S. Katti, and D. Katabi. Trading structure for randomness in wireless opportunistic routing. In ACM SIGCOMM, 2007.

[6] Christos Gkantsidis and Pablo Rodriguez. Network coding for large scale content distribution. In IEEE INFOCOM, 2005.

[7] D. S. J. De Couto, D. Aguayo, J, Bicket, and R. Morris. A highthroughput path metric for multi - hop wireless routing. In ACM MobiCom' 03, San Diego, California, September 2003.

[8] C. Fragouli, J. WIdmer, and J. LeBoudec. Efficient broadcasting using network coding. IEEE/IACM Transactions on Networking, 2007.

[9] S. Gollakota and D. Katabi. ZigZag decoding: Combating hidden terminals in wireless networks. In SIGCOMM, 2008.

[10] T. Ho, R. Koetter, M. Medard, M. Effros, J. Shi, and D. Karger. Arandom linear network coding approach to multicast. IEEE Transactions on Information Theory, 52 (10), 4413 - 4430, October 2006.

[11] S. Jaggi, M. Langberg, S. Katti, T. Ho, D. Katabi, M. Medard, and M. Effros. Resilient network coding in the presence of byzantine adversaries. IEEE Transactions on Information Theory, 2006.

[12] S. Jaggi, P. Sanders, P. A. Chou, M. Effros, S. Egner, K. Jain, and L. Tolhuizen. Polynomial time algorithms for multicast network code construction. IEEE Transactions on Information Theory, 2003.

[13] A. Jiang. Network coding for joint storage and transmission with minimum cost. In ISIT, 2006.

[14] S. Katti, S. Gollakota, and D. Katabi. Embracing wireless interference: Analog network coding. In ACM SIGCOMM, 2007.

[15] S. Katti, D. Katabi, H. Balakrishnan, and M. Medard. Symbol - level network coding for wireless mesh neworks. In ACM SIGCOMM, 2008.

[16] S. Katti, D. Katabi, W. Hu, H. S. Rahul, and M. Medard. The importance of being opportunistic: Practical network coding for wireless environments, Johns Hopkins University, Department of Computer Science, 2005.

[17] S. Katti, H. Rahul, D. Katabi, W. Hu, M. Medard, and J. Crowcroft. XORs in the air: Practical wireless network coding. In ACM SIGCOMM, 2006.

[18] R. Koetter and M. Medard. An algebraic approach to network coding. IEEE/ACM Transactions on Networking, 2003.

[19] S. R. Li, R. W. Yeung, and N. Cai. Linear network coding. In IEEE Transactions on Information Theory, 2003.

[20] J. Liu, D. Goeckel, and D. Towsley. Bounds on the throughput gain of network coding in unicast and multicast wireless networks. IEEE J. Sel. A. Commun., 27 (5), 2009.

[21] D. S. Lun, M. Medard, and R. Koetter. Efficient operation of wireless packet networks using network coding. In International Workshop on Convergent Technologies (IWCT), 2005.

[22] A. K. Miu, H. Balakrishnan, and C. E. Koksal. Improving Loss Resilience with Multi - Radio Diversity

in Wireless Networks. In 11th ACMMOBICOM Conference, Cologne, Germany, September, 2005.

[23] C. Reis, R. Mahajan, M. Rodrig, D. Wetherall, and J. ZahOljan. Measurement - based models of delivery and interference. In SIGCOMM, 2006.

[24] S. Zhang, S. Liew, and P. Lam. Physical layer network coding. In Proc. of ACM MOBICOM 2006, Los Angeles, USA.

[25] S. Sengupta, S. Rayanchu, and S. Banerjee. An analysis of wireless network coding for unicast sessions: The case for coding - aware routing. In Proc. of IEEE INFOCOM, pages 1028 - 1036, 2007.

[26] J. Widmer and J - Y. L. Boudec. Network coding for efficient communication in extreme networks. In ACM SIGCOMM Workshop on DTNs, 2005.

[27] Y. Wu, P. A. Chou, and S. Y. Kung. Information exchange in wireless networks with network coding and physical - layer broadcast. MSR - TR - 2004 - 78.

[28] Y. Wu, P. A. Chou, and S. - Y. Kung. Minimum - energy multicast in mobile ad hoc networks using network coding. IEEE Transactions on Communications, 2007.

第3章　利用网络编码实现P2P网络的内容分发和多媒体流

Chen Feng、Baochun Li

加拿大，多伦多，多伦多大学电子与计算机工程系

摘要

对等网络（P2P）已经成为实现网络编码潜力的最有前途的平台之一，因为在互联网边缘处的终端主机（简称为对等体）上有丰富的计算资源与现代的处理器。本章中，我们将探寻网络编码在P2P网络中的应用，并专注于其中两个重要的应用：内容分发和多媒体流。对于每个应用，我们都运用网络编码技术探讨了可能的P2P系统的设计空间，并直观地说明了网络编码技术的优点。此外，我们还进一步讨论了几个理论成果和实践问题。

关键词

对等网络、内容分发、多媒体流、网络编码、广播。

3.1　运用网络编码的内容分发

P2P内容分发已成为现代越来越普遍的一个内容分发协议，该协议非常简单。考虑单个服务器分发一个较大的文件（通常是几百兆甚至千兆字节）给互联网上的很多终端主机的情况（对等体）。服务器并不是给每一个单独的对等体上传文件，而是首先将文件分为 r 个数据块，然后通过让对等体们彼此交换信息实现对这些数据块的有效分发。

P2P内容分发的基本优点是显著地减少每个对等体的文件下载时间。直观地说，参加传输的对等体将贡献自己的上载带宽来为其他对等体服务，使得系统中的总上载带宽显著增加，从而实现更快的文件分发过程。

3.1.1　怎样在P2P内容分发中使用网络编码?

BitTorrent[1]已经演变成当下最流行的P2P内容分发协议，参考文献［2］中详细描述了BitTorrent系统。在BitTorrent系统中，新加入该系统的对等体应联系中心跟踪器，以获得参与当前文件下载的对等体名单。通常情况下，跟踪器从参与下载的所有对等体中随机提供50个给新加入者，然后新的对等体将和约40个对等体尝试建立和保持联系，并成为邻居。如果一个对等体没有成功保持至少20个邻居（比如由于对等体离开等原因），它就会再次联系跟踪器以获得额外的节点列表。在这种方式下，我们称由所有参与当前下载的对等体组成的动态逻辑网络为P2P覆盖网络。

参与 BitTorrent 的所有对等体合作使用蜂拥技术下载文件。待下载的文件被划分为 r 个大小相等的数据块 $\{b_1, \cdots, b_r\}$。每个对等体都与邻居交换数据块，直到获得所有的 r 个数据块便可以离开系统。需要注意的是，服务器也可以看做是邻居。对等体在下载完新的数据块后会通知它所有的邻居，因此每一个对等体都知道它的邻居中可利用的数据块信息。当对等体从某个特定的邻居请求一个数据块时，它通常会请求本地最稀缺的数据块，也就是在其所有邻居中最不常见的数据块。这样做的目的是确保数据块能够几乎均匀地在覆盖网络中传播，否则这些在系统中非常稀缺的数据块将成为信息瓶颈，可能会导致对等体的下载时间大大增加。

一般来说，所有 P2P 内容分发系统中的参与对等体在请求数据块时都需要回答以下问题：应该下载哪些数据块，以及从哪些邻居下载？这些问题被称为数据块调度问题，我们需要通过一个只利用本地信息且有效的分布式方式来解决这些问题。BitTorrent 采用本地稀有优先策略来避免信息瓶颈。然而，我们可以借助于网络编码以一种简单而有效的方式来解决上述问题。

Avalanche[3, 4] 就是这样一个使用了网络编码的 P2P 内容分发协议。在 Avalanche 中，待下载的文件再次被划分成 r 个等长的数据块，每个数据块 b_i（$i=1, \cdots, r$）都可以看做是有限域 $\mathbb{F}_q$ 上的一个长度固定的向量。对等体（或服务器）将编码块而不是原始数据块上传给它的邻居，其中每个编码块都是对等体已经收到的编码块的一个随机线性组合。随着这个改变的发生，系统中的对等体没有必要再请求特定的数据块。相反，系统中的对等体可以随意地从它的邻居那里下载编码块，直到它能够重建原始文件。

下面用一个简单的例子[5] 来更好地说明系统是如何工作的。在文件分发过程的开始，对等体（如对等体 A）联系服务器并接收一定数量的编码块。例如，服务器上传两个编码块 x_1 和 x_2 给对等体 A，其中

$$x_i = \alpha_1^i b_1 + \alpha_2^i b_2 + \cdots + \alpha_r^i b_r,\ i = 1, 2 \tag{3-1}$$

式中的 α_j^i（$1 \leqslant j \leqslant r$）随机选取自域 $\mathbb{F}_q$。当对等体 A 需要为相邻对等体（如对等体 B）提供服务时，它将简单地生成一个编码块 x_3

$$x_3 = \alpha_1^3 x_1 + \alpha_2^3 x_2 \tag{3-2}$$

其中，α_1^3 和 α_2^3 也随机选取自域 $\mathbb{F}_q$。将式（3-1）代入式（3-2）可以得到

$$x_3 = \sum_{j=1}^{r} (\alpha_1^3 \alpha_j^1 + \alpha_2^3 \alpha_j^2) b_j$$

这意味着 x_3（可推广到系统中的所有编码块）是原始数据块 $\{b_1, \cdots, b_r\}$ 的某种随机线性组合，相应的向量（$\alpha_1^3\alpha_1^1 + \alpha_2^3\alpha_1^2, \cdots, \alpha_1^3\alpha_r^1 + \alpha_2^3\alpha_r^2$）被称为 x_3 的全局编码向量。继续上述过程，系统中的每个对等体能够收集 r 个线性无关的编码块。那么，对等体怎样才能从这些编码块中恢复出原始文件呢？正如在第 1 章中说明的，我们会把全局编码向量作为"报头"添加到每个数据块中，对等体在接收到 r 个线性无关的编码块后就可以利用这一信息来重建原始文件。

3.1.2　为什么网络编码有助于 P2P 内容分发？

网络编码为 P2P 内容分发带来的潜在好处是什么？回想一下，P2P 内容分发的难点

在于找到一个最优的数据块调度算法，该算法应能够通过分布式的方式尽量减少文件下载的时间。在覆盖网络实时动态变化的情况下，这个问题变得更具有挑战性。

不使用网络编码时，每个节点都需要决定从哪个邻居那里下载哪些数据块，并且只能根据本地信息来判断。对全局视野的缺乏导致节点选择的往往不是最优解决方案，因为本地最稀有的数据块可能不是全局最稀有的。与此相反，当我们使用网络编码时，所有编码块对于对等体的价值都几乎相同，我们不再需要在系统中定位并请求全局最稀有的块，因此避免了信息瓶颈，并且反过来减少了文件的下载时间。

网络编码的另一个重要优点是对于对等体的离开具有鲁棒性。如果没有网络编码，很有可能一些数据块会因为服务器的离开或因为拥有这些数据块的对等体的离开而丢失。当这个不幸的事件发生时，剩余的对等体将不能恢复出原文件。另一方面，使用网络编码时，由某些数据块的丢失导致的风险便不再是问题了。直观地说，网络编码对数据块的混合使得每一个数据块都分散到了系统中大量的编码块中。

总之，网络编码以一种令人惊讶的简单而有效的方式解决了关于块调度的问题，缩短了文件的下载时间。此外，它还为对等体的离开提供了理想的鲁棒性，人们不用再担心丢失某些数据块了。所有这些都有助于网络编码在 P2P 内容分发中发挥作用。接下来将给出一些理论成果，以证实网络编码的这些优势。

3.1.3 利用网络编码的 P2P 内容分发的理论结果

下面从 P2P 内容分发系统模型开始讨论。覆盖网络由单台服务器和一些参与对等体组成，可以用有向图 $G=(V, E)$ 来表示，其中 $s\in V$ 表示服务器，顶点 $v\in V-\{s\}$ 表示一个参与对等体，边 $e\in E$ 表示从一个对等体到另一个对等体的一个覆盖连接[⊖]。

在没有对等体加入或离开的情况下，该模型可以很好地适用于静态 P2P 内容分发系统。然而在文件分发的过程中，对等体可能在任意时间加入或离开系统。我们可以采用如下网格图形技术[5, 6]来模拟数据块传输以及这种对等体的动态变化。对于有向图 $G=(V, E)$，构建一个新的网格图 $G^*=(V^*, E^*)$，其节点集为

$$V^* = \{(i, t): i \in V \text{ 且 } t \in \{t_0, t_1, t_2, \cdots\}\}$$

其中，节点 $(i, t) \in V^*$ 表示 t 时刻的节点 $i\in V$，集合 $\{t_0, t_1, t_2, \cdots\}$ 表示对等体动态事件和块传输事件的所有起始时间和结束时间。边集 E^* 通过服务器的策略以及所有其他在集合 V 中请求并上传编码数据块的节点来确定。具体而言，网格图中有两种类型的边：

1）如果节点 i 上载了一个编码块给节点 j，该上载过程开始于时刻 t_k，结束于时刻 t_l，那么我们就在网格图中从节点 $(i, t_k) \in V^*$ 添加一条到节点 $(j, t_l) \in V^*$ 的边 $e\in E^*$。这种类型的边表示邻居对等体之间的编码块的传输。

⊖ 如果对等体 i 与另一个对等体 j 保持 TCP 连接，我们就说从 i 到 j 有一条边。注意，G 的边不是物理通信链路，而是由包含若干物理链路的路径组成的抽象连接。

2）假设节点 i 在时刻 t_k 加入系统，在时刻 t_l 离开系统。对于所有满足 $k \leqslant m \leqslant l-1$ 的 m，都有一条从节点（i，t_m）到节点（i，t_{m+1}）的无限容量的边。这种类型的边表示随着时间的推移在节点 i 接收到的信息的积累。

这里提出的网格图模型是对参考文献［5，6］中采用的模型的概括。图3-1所示的是时间范围为 $t \leqslant t_8$ 的网格图 G^*，其中容量为∞的边被高亮显示。需要注意的是，无论有向图 G 是否有环，网格图 G^* 始终是无环的。下面的分析是对参考文献［5］的直接延伸。

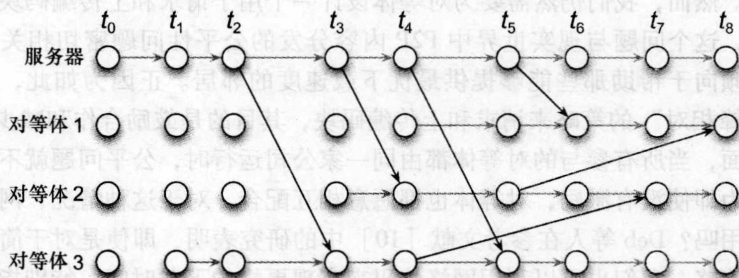


图3-1　网格图 G^* 示例

在时刻 t_0，服务器 s 准备通过蜂拥与网络编码技术分发由 r 个数据块组成的文件，如3.1.1节所述。这可以被精确建模为节点$(s, t_0) \in V^*$通过随机线性网络编码将 r 个数据块多播给 G^* 中的所有的其他节点。

我们现在来为某个特定的节点 $i \in V$ 确定下载时间。考虑网格 G^* 中从（s，t_0）$\in V^*$ 到节点（i，t_l）$\in V^*$ 的最大流，用 $f_i(t_l)$ 表示。当 $f_i(t_l) \geqslant r$ 时，节点（i，t_l）在随机线性网络编码下可以恢复出整个文件的概率接近于1，前提是域的大小 q 足够大（有关详细信息，请参阅第1章中的讨论）。令 $m(i)$ 表示使得 $f_i(t_l) \geqslant r$ 成立的 l 的最小值，即 $m(i) = \min\{l: f_i(t_l) \geqslant r\}$，则节点 $i \in V$ 的下载时间等于 $t_{m(i)}$ 的概率很大。需要注意的是，根据信息论的割集约束，$t_{m(i)}$ 也是下载时间的下限。这意味着，利用随机线性网络编码系统可以实现其最短下载时间。即使罕见的事件确实发生，比如节点 $i \in V$ 由于线性相关不能在时刻 $t_{m(i)}$ 重建原始文件，它在收到一些额外的编码块后最终也可以恢复出文件。通过使用一个简单的概率论据可以证明这一点。

上述分析显示了网络编码的第一大优势——对等体不需要决定下载哪个数据块，同时对于任何给定的请求或编码块的上传策略还能实现最低的下载时间。下面研究网络编码的第二大优势——对于对等体离开系统的鲁棒性。

当某些对等体甚至是服务器离开系统时，我们很自然地会关注剩余的对等体能否重建原始文件并完成下载。这个重要的问题可以再次通过使用上述的网格图技术来回答。对于任意给定的时刻 t_l，假设我们感兴趣的是在没有更多的节点在时刻 t_l 之后离开系统的前提下，系统中剩余的对等体是否可以恢复出原始文件。为此，令 $R(t_l) = \{(i, t_l): i \in V\}$，并令 $f_R(t_l)$表示网格图中从节点$(s, t_0) \in V^*$到节点集 $R(t_l)$的最大流量。那么当

$f_R(t_l) \geq r$ 时，剩余对等体可以恢复出整个文件的概率很大[7]。请注意，当 $f_R(t_l) < r$ 时，根据信息论的割集约束，即使剩余的对等体已知全局信息，它们也不可能恢复出整个文件。这意味着使用网络编码确实能够获得最优的鲁棒性。对于在需要通过不可靠对等体提供可靠数据访问的 P2P 存储系统中存在的鲁棒性问题，参考文献［8，9］中的研究表明，使用网络编码也可以获得类似的好处。

因此看到，相比于对编码块的请求和上传策略，在 P2P 内容分发中使用网络编码策略能够同时实现最小下载时间和最大鲁棒性。对于给定的策略，这几乎是可以预期的最好的性能。然而，我们仍然需要为对等体设计一个用于请求和上传编码块的良好的策略。实际上，这个问题与现实世界中 P2P 内容分发的公平性问题密切相关，因为对等体会自然地倾向于帮助那些能够提供最优下载速度的邻居。正因为如此，Avalanche[4] 采用了“针锋相对”的策略来请求和上传编码块，其目的是鼓励合作和减少搭便车。

另一方面，当所有参与的对等体都由同一家公司运行时，公平问题就不再是什么大问题了，因为即使没有激励，对等体也都愿意相互配合。对于这种情况，网络编码可以起到帮助作用吗？Deb 等人在参考文献［10］中的研究表明，即使是对于简单的编码块请求和上传策略，我们也可以利用网络编码来实现更短的下载时间。特别指出的是，他们假设系统模型开始由 n 个参与的对等体和 r 个数据块组成。最初，每个对等体只有 r 个数据块中的一个，并且每个数据块都被均匀地分散在系统中。系统的目标是尽可能快地向所有的对等体分发所有 r 个数据块。为了简单起见，他们假定时间被分成时隙并且各个对等体的时隙是同步的。在每个时隙期间，每个对等体向所有的邻居节点请求编码块，然后每个对等体随机均匀地从那些发送请求的邻居中选择出一个邻居，并利用随机线性网络编码上传一个编码块。也就是说，对等体在每个时隙中简单地采用一个随机策略上传一个编码块。下面的定理给出了使用网络编码时这一策略的性能。

定理 3-1[10] 假设基础覆盖网络完全连接。假设域的大小 $q \geq \max\{r, \ln(n)\}$。令随机变量 T_b 表示上述系统模型和策略下的广播时间（所有对等体下载所有数据块所需的时间），则

$$T_b \leq 5.96r + O(\sqrt{r\ln(r)}\ln(n))\text{，概率为 } 1 - O\left(\frac{1}{n}\right)$$

此外，令 T_i 表示对等体 i 的下载时间，则

$$E[T]_i \leq 5.96r + O(\sqrt{r\ln(r)}\ln(n))$$

观察到当 $r = \Theta(n)$ 时，所有对等体至少需要 $\Theta(n)$ 个时隙才能下载完所有数据块。因此，从定理 3-1 可以看出，这个简单的利用网络编码的策略对于 $r = \Theta(n)$ 是最优的。他们还可以证明，如果没有网络编码，这个简单的策略反而会恶化系统在广播时间方面的性能。也就是说，在充分合作的环境中，当 $r = \Theta(n)$ 时，使用网络编码可以提高系统性能。此外，他们的仿真结果还表明，当 r 较小时网络编码的好处仍然存在。

以往的理论研究结果表明，使用网络编码有助于减少广播时间，这个广播时间指的是所有对等体的最大下载时间。于是我们自然要问：那么网络编码对其他关于下载时间的函数的作用是怎样的呢？比如网络编码是否有助于减少平均下载时间？事实上，正如

参考文献［11］中指出的，当给定对等体完成下载的顺序时，使用网络编码可以实现最优延迟区域中的任意点，尤其是平均下载时间。参考文献［12］将该结果部分延伸到动态变化的网络场景中，其研究表明网络编码提供了一个优于路由的鲁棒性方案。

我们已经给出了许多网络编码在 P2P 内容分发中的优势，然而这些优势不是没有代价的。事实上，与传统方法相比，网络编码需要额外的计算资源。接下来的问题自然就出现了：我们可以在没有显著性能损失的情况下减少计算成本吗？

Chou 等人在参考文献［13］中提出了分组网络编码这一概念，即一个文件被分成大小相等的段（也称为代），每段又被进一步划分成大小相等的数据块。该方法对同一段内的数据块进行编码操作，但不对不同段之间的数据块进行编码。虽然分组网络编码可以在很大程度上减少计算复杂度，但乍看之下它在文件下载时间和对于对等体离开的鲁棒性方面的作用并不明显。

参考文献［14］中证明了使用分组网络编码仍然能以很高的概率为任意给定的编码块请求和下载策略实现最小的文件下载时间。此外，采用和 Raptor 码中类似的预编码技术（详见参考文献［14］）来解码分组网络码可以进一步减少计算成本。也就是说，在不明显损失下载时间性能的条件下可以节省大量的计算操作。

我们接下来研究分组网络编码在对等体离开的鲁棒性方面的影响。直观地说，随着分段数量的增加，每个分段包含的数据块数量会减少，则由于对等体的离开而导致某个分段的所有数据块都丢失的概率会增加，最终导致鲁棒性降低。另一方面，较少的分段带来了高鲁棒性，但会减少对计算成本的节省。换句话说，对于 P2P 内容分发中的网络编码来说，存在着一个鲁棒性和复杂度之间的折衷关系。如果能够工作在一个“最佳点”，即在计算成本合理的基础上获得最优的鲁棒性，这就再好不过了。在参考文献［15］中，这种鲁棒性 - 复杂度的折衷关系被定量表征出来，为选择这样的“最佳点”提供了理论指导方针。

总之，网络编码不需要复杂的块调度算法就能最佳地利用现有的覆盖连接。同时，即使大量对等体突然离开系统，它也能提供可能的最佳鲁棒性。此外，如果是对每个分段而不是整个文件进行编码操作，那么我们可以大大节省计算成本，同时还没有显著的性能损失。

3.1.4　网络编码实现 P2P 内容分发的实际应用

网络编码的成本主要是计算成本，这可能会需要过高的计算资源。正如 3.1.3 节中讨论的，使用分组网络编码可以在没有明显的性能损失的条件下减少计算成本。然而，在复杂度和性能之间一般存在一个权衡。因此，与现实利益相关的问题可能是：利用现代的处理器，我们所能负担的最大计算成本是多少？在实际中可能负担的数据块的数量和大小是多少？参考文献［16］通过在应用层高性能地实现随机线性网络编码，已经解决了以上问题。参考文献［16］中给出了许多重要的研究结果，以下是一个简短的总结。

1）为了保证适度的编码和解码，各分段包含的数据块的数量应小于几百个，数据

块的大小应小于几百兆。

2）各分段包含的数据块数量对性能的影响比数据块大小的影响更大。这表明，我们应为每个分段分配数目较少的数据块（如100个）。

3）最佳的数据块大小约是2～32KB，这能保证最快的编码和解码。这个最佳值随着分段中包含的数据块数量的增加而增加。

事实上，Avalanche中的每个分段包含80个数据块，这与上述第二个研究结果相符。Avalanche中的每个数据块约为2.3MB，依据上述第一个研究结果这是合理的。为什么Avalanche不进一步减小数据块的大小呢？一部分原因是由于参与对等体之间的连接较慢。当连接速度较慢时，上传和下载速度成为瓶颈，编码和解码的速度反倒影响较小。此外，较大的数据块意味着更少的"报头"开销，这在实际中是我们希望看到的。

关于网络编码的另一个实际问题是抵挡恶意节点的干扰。事实上，即使只有一个损坏的数据块——由恶意对等体注入网络中——也有可能污染大量的编码块，并阻碍参与对等体的解码。那么，我们可以发现并改正损坏的编码数据块吗？Avalanche中采用了安全随机校验[4]，以较低的计算成本对损坏的编码块进行实时检测。另一种解决恶意攻击的方案是利用网络纠错码[17]，这已经成为了一个新的基于网络编码的研究领域。

3.2 运用网络编码的P2P多媒体流

近年来，因为要为数以百万计的现实世界中的用户应用程序提供实时的流媒体内容，导致互联网上的对等（P2P）多媒体流出现了前所未有的增长。P2P流媒体的核心优势是大幅增加在有专用流媒体服务器的流媒体通道中可以维持的对等体数量。直观地说，如果参与对等体互相贡献自己的上传带宽给相同信道中的其他对等体，那么专用流媒体服务器上的负载就能得到明显的缓解。

有一些基本的性能指标可以用于表征一个"良好的"P2P流媒体系统。让我们来看看其中的几个例子。在播放质量方面，如果流媒体的内容没有及时到达，它在播放时就必须被跳过，这会导致播放质量下降。怎样才能使所有的参与对等体都始终保持较高的播放质量呢？在初始缓冲延迟方面，当一个对等体第一次连接或切换到一个新的流通道时，需要如何提高用户体验，提供最短的缓冲延迟？在服务器带宽成本方面，如何通过使参与对等体最大限度地提高带宽贡献来减少此类成本？最后，如何设计一个能很好地进行扩展的系统，以适应大型的"瞬间拥塞"和对等体的高度动态化？

当评估一项专为P2P多媒体流设计的协议时，这些性能指标应被优先考虑。播放质量和初始缓冲延迟对于用户体验最为重要，它决定了用户的满意度水平。然而，服务器带宽的成本对于运营公司最为重要，因为它们直接决定了绝大部分的持续运营成本。下面，我们将演示如何使用网络编码设计出满足所有这些指标的P2P多媒体流系统。

3.2.1 网络编码怎样应用于P2P多媒体流？

目前已经有大量P2P多媒体流协议被提出，从策略上来说，它们一般被分为两类。

树形 push 流策略（例如参考文献［18］）将参与对等体组成一个或多个组播树，并沿这些树间分发流媒体内容。与此相反，网状 pull 流策略将对等体组成一个随机的网状结构，每个对等体都有一个由若干参与对等体组成的随机子集作为其邻居。流处理的内容被分成一系列的数据块，每个数据块都是一段很短的播放内容。每个对等体都会维护一个由即将被播放的数据块组成的播放缓冲区。所有对等体都会定期与它的邻居交流播放缓冲区（通常称为缓存映像）中的数据块可利用信息。基于这些信息，数据块将从适当的邻居那里被拖取，以满足其播放的截止期限。未被及时接收到的数据块将在播放过程中被跳过，并导致播放质量下降。

组播树的建立和维持在实践中相对困难，尤其是当对等体频繁地加入或离开系统的时候。相比于树形 push 流策略，网状 pull 流策略不需要建立和维持组播树，这使得它在处理动态对等体时展现出了很好的简单性和鲁棒性，而这实际上是继承了类似比特流内容分发系统的设计理念。出于这个原因，大多数现实世界中的 P2P 多媒体流系统，如 CoolStreaming[19] 和 PPLive[20]，采用的都是网状 pull 策略。

在这样的流媒体系统中，每个参与的对等体都需要决定下载哪个数据块以及从哪个邻居那里下载。这个问题与比特流系统中的数据块调度问题是类似的，但目的却截然不同。这里的目的并不是实现文件下载时间的最小化，而是最大限度地减少播放跳跃间隙，从而提高播放质量。对于新加入的对等体，一个较短的初始缓冲延迟也是块调度算法设计中的重要关注点。

受到网络编码对内容分发系统中的块调度问题的成功解决的启发，人们可能会猜想类似的方法也可以很好地应用于多媒体流系统。事实上，Wang、Li 等人[21] 通过在实验测试平台上用分组网络编码方案取代传统的块调度算法，评估了网络编码应用于 P2P 多媒体流的有效性。研究发现，当整体带宽供应勉强超过需求时，或当对等体的到来和离开不稳定时，网络编码可以提供一些边际效益。

由于使用网络编码带来的负面结果是如此的微不足道，因此人们可能会认为，只在传统的网状 pull 流策略下，简单地更换数据块调度算法可能无法完全发掘网络编码的优势。这激发了对基于网络编码的 P2P 多媒体流技术的重新设计。事实上，Wang 和 Li 在参考文献［22］中提出了 R^2——一个从头开始设计的充分利用网络编码的流媒体算法。下面将简要概述 R^2 的设计原则。

3.2.1.1 随机网状结构中的随机 push

在传统的网状 pull 流协议中（为简便起见，以下简称 PULL），流媒体传输的内容按顺序被分成大小相等的数据块序列。在 R^2 中，流媒体传输的内容先被分成大小相等的分段序列，每个分段又被进一步划分成 k 个大小相等的数据块。编码操作仅在各分段内部进行，而不在不同的分段之间进行。这样做的目的是为了降低计算成本。

假设一个对等体 p 已经接收到了 m 个($m \leq k$)分段 s 中的编码块，记为 $[b_1, b_2, \cdots, b_m]$。当对等体 p 需要为其下游的对等体提供分段 s 时，它会生成一个编码块 x，如图 3-2 所示。

$$x = \alpha_1 b_1 + \alpha_2 b_2 + \cdots + \alpha_m b_m$$

其中，$\alpha_j(1\leqslant j\leqslant m)$随机选取自有限域$\mathbb{F}_q$。正如3.1.1节中指出的，$x$最终可以表示成为分段$s$中原始块的随机线性组合。因此$p$的下游对等体只要接收到$k$个线性独立的编码块就可以解码得到分段$s$。

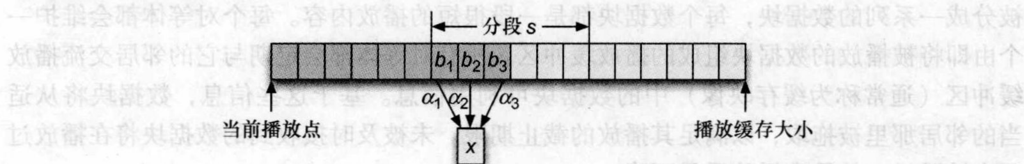


图3-2　该示例说明了对等体p上的编码操作，其中对等体p收到了分段s中的3个编码块，每个分段由6个编码块组成

在PULL中，对等体会向合适的邻居请求并拖取它丢失的数据块，如果在给定时间内它还没有接收到该数据块（例如由于对等体离开等原因），则它必须冒着错过截止期限的风险再次请求并拖取该数据块。而在使用网络编码的R^2中，多个邻居可以同时提供对等体丢失的数据分段，并且不需要任何明确协调，如图3-3所示。使用这种方法时，参与的对等体能够在一个随机网状结构上执行push操作（即上游对等体无需接收任何来自下游对等体的请求，就可以决定提供哪些分段），从而充分利用现有的覆盖连接。更重要的是，即使几个邻居突然离开系统，如果其他提供同样分段的邻居仍然在工作，那么错过截止期限的概率就很小。

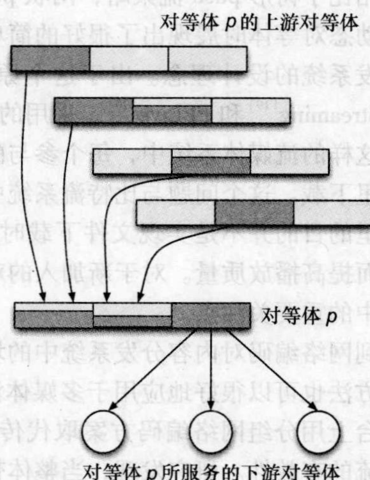


图3-3　随机网状结构上的随机push示例。利用网络编码，多个上游对等体能够在没有任何明确协调的情况下对同一分段内的编码块进行push操作

3.2.1.2　下游对等体的及时反馈

在push编码块之前，上游对等体应随时获得有关其下游对等体丢失的分段的精确信息，这需要参与的对等体及时在系统中交流它们的缓存映像。为了避免过多的开销，在PULL中是定期交流缓存映像，但是R^2可以负担得起“实时”交流缓存映像——每当一个对等体播放或接收完一个分段时，它就会向它的所有邻居发送一个新的缓存映像。

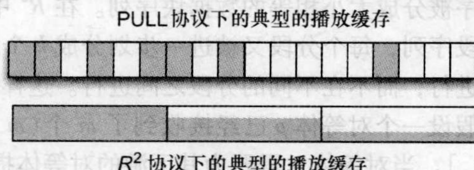


图3-4　R^2和PULL的缓存映像比较

为什么R^2能够这样“实时”地交流缓存映像呢？由于R^2使用的是较大的数据分段而不是较小的数据

块（见图 3-4），其缓存映像指示的是分段可用信息，而不是数据块可用信息，因此 R^2 的缓存映像可以小一个数量级。此外，当使用较大的分段时，播放或完成下载一个分段需要更长的时间，这使得对缓存映像的更新较慢。因此在合理的开销范围内，R^2 可以实现缓存映像的状态一旦更新就立刻被发送给所有邻居。

3.2.1.3　同步播放和初始缓冲延迟

使用更大的分段还更易于不同参与对等体之间的播放缓冲同步，这使得所有对等体都在几乎相同的时间播放相同的分段。R^2 具有如下的同步播放特性：当一个对等体加入或切换到一个新的流媒体通道时，它首先从它的邻居处取回缓存映像，同时也取回当前播放的分段信息。要同步播放缓冲，新来的对等体会跳过几个分段，并只获取位于当前播放点之后 D 秒的分段，这里的 D 可以看做是初始缓冲延迟，它取决于当前的播放点和被跳过的分段的数量（见图 3-5）。无论播放缓冲区的状态如何，D 秒之后该对等体都将准时开始播放。

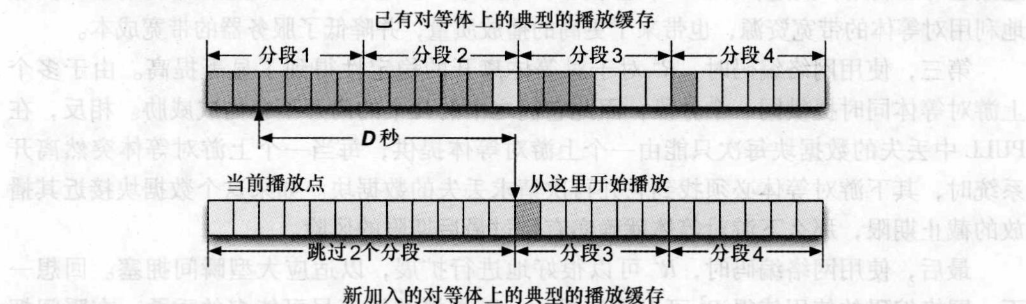


图 3-5　R^2 的初始缓冲延迟示例。可以看出新加入的对等体的初始缓冲延迟由当前的播放点和被跳过的分段的数量所决定

当播放同步时，对等体的播放缓冲器之间可以尽可能地重叠，因此对等体能够更有效地互相帮助。这一理想特点在瞬间拥塞的情况下特别重要，这时有大量的对等体大约在同一时间加入流媒体通道。由于这些新来的对等体请求的分段都差不多相同，因此一旦获得了少量的编码块它们就能够互相帮助。这将使它们能够更好地利用上传带宽，并反过来又提高了流媒体系统的可扩展性。

最后要特别指出的是，R^2 代表了一个简单的设计理念，而不是一个严格的协议设计。R^2 的设计空间是灵活的，可以容纳更多精心设计的用于不同目的的协议。例如，R^2 中的对等体有充分的自由来决定要把哪些分段 push 给哪些邻居。push 策略在做这个决定时，可以考虑时序要求以确保对重要的分段的顺利播放，考虑公平性问题以鼓励合作和减少搭便车，以及考虑地理因素以减少不同的互联网服务供应商之间的流量。

3.2.2　为什么网络编码对多媒体流有帮助？

多媒体流应用对时序的要求十分严格，这也是它与内容分发应用程序的一个重要区别。这样的约束使得网络编码的优势不是太明显。事实上，参考文献［21］中证明了网

络编码在内容分发中的成功应用不能被简单地复制到多媒体流中。若要充分利用网络编码，需要对 P2P 流算法进行一个完整的重新设计。如前所述，这促使了对 R^2 的设计和实现。本节中，我们将直观地解释为什么在 R^2 中使用网络编码能够为流媒体系统提供良好的整体性能。

首先，与 PULL 中的典型数据块相比，使用网络编码的 R^2 允许更大的分段。使用更大的分段可以实现对缓存映像的"实时"交流，并且不会带来额外的开销（甚至还可以减少开销）。通过 R^2 对缓存映像的实时更新，参与的对等体能够更好地相互服务，这与 PULL 中以周期的方式交流缓存映像形成了鲜明的对比。参考文献［23，24］中的研究表明，缺乏对缓存映像的及时交流可能是 PULL 的实际表现并非最优的一个主要因素。

其次，使用网络编码时，R^2 中的对等体执行的是 push 而不是 pull 操作，这使得它们可以更好地利用它们的上传带宽资源。更重要的是，在 R^2 中，即使是速度缓慢的覆盖连接也可以利用，而这在 PULL 中通常是不可能的[23]。总之，所有这些因素都有助于更好地利用对等体的带宽资源，也带来了更高的播放质量，并降低了服务器的带宽成本。

第三，使用网络编码时，R^2 对于对等体离开的稳定性得到了显著提高。由于多个上游对等体同时提供同一个分段，因此它们之中的几个的离开不会构成威胁。相反，在 PULL 中丢失的数据块每次只能由一个上游对等体提供，每当一个上游对等体突然离开系统时，其下游对等体必须找到它并再次请求丢失的数据块。如果这个数据块接近其播放的截止期限，那么下游对等体就确实有错过最后期限的风险。

最后，使用网络编码时，R^2 可以很好地进行扩展，以适应大型瞬间拥塞。回想一下，网络编码的使用使得 R^2 可以同步播放。随着播放缓存尽可能多的重叠，在瞬间拥塞期间，新来的对等体在收到了一些编码块后能够立即互相帮助。这可以充分利用新来的对等体的上载带宽，并反过来大大提高了流系统的可扩展性。

表 3-1 系统模型中的关键符号

U_i	一个第 i 级对等体的上传能力（数据块/s）
U_p	参与对等体的平均上传能力
U_s	服务器上传能力（数据块/s）
R	流速率（数据块/s）
D	初始缓冲延迟（s）
N	对瞬时拥塞的扩展能力（系统中参与对等体的数量）
k	每个分段中包含的数据块数量
δ	服务器能力$\left(=\frac{U_s}{NU_p}\right)$

3.2.3 基于网络编码的 P2P 多媒体流的理论结果

本节中，我们给出了一系列关于基于网络编码的 P2P 流媒体系统的性能分析结果，并专注于 R^2 的基本限制和可实现的性能。为了便于数学处理，我们对系统模型做了一

些假设。表 3-1 中总结了下文涉及的关键符号，以便于查询。

首先，根据对现有的 P2P 系统的测量研究（例如参考文献［25］），我们假定对等体的上传能力是流媒体系统中唯一的瓶颈。其次，为了表征对等体上传能力的非均匀性，我们采用了参考文献［26］中的两级模型，该模型将系统中的对等体大致分为两级，每级对等体的上传能力大致相同㊀。我们使用 U_p 表示参与对等体的平均上传能力，U_s 表示一个专用流媒体服务器的上传能力（当有多个流媒体服务器存在时，它们可以被看做是一个虚拟的“超级服务器”）。

下面我们讨论基于网络编码的 P2P 流媒体系统的一些基本的性能极限。首先，我们发现总带宽消耗不应大于总带宽供应。这将引出下面的定理，该定理已经在参考文献［26］中被证明。

定理 3-2[26]　最大流速率 $R_{\max}$ 的计算公式如下：

$$R_{\max} = U_p + \frac{U_s}{N}$$

其中，N 是系统中参与对等体的数量。

接下来考虑系统中所有的参与对等体缓冲一个分段所需的缓冲延迟。一方面，因为一个分段由 k 个数据块组成，因此要使 N 个对等体都缓冲一个分段，至少需要传输 kN 个数据块。另一方面，考虑到只有在对等体获得这一分段中的至少一个编码块后，它才能提供这个分段，因此总上传速率的上界是 $U_s + NU_p$。从而我们可以对缓冲延迟作出以下约束：

定理 3-3　令随机变量 D_s 表示一个分段的缓冲延迟，在上述系统模型下，对于任意给定的 push 策略有：

$$E[D_s] \geqslant \frac{kN}{U_s + NU_p}$$

其中，N 是系统中参与对等体的数量；k 是每个分段中数据块的数量。

需要注意的是，为了保证播放流畅，至少有一个分段应该被缓冲。由定理 3-3 可知，参与对等体要实现这一点所需的时间的期望值至少是 $E[D_s]$ 秒。换句话说，定理 3-3 给出了一个瞬时拥塞情况下的最短初始缓冲延迟的下限。

鉴于上述性能限制，我们需要回答以下两个问题：

- R^2 实现良好的整体性能的充分条件是什么？
- R^2 的性能距离最优还有多远？

这些问题对于理解 R^2 的基本性质和局限性至关重要。这里我们主要研究瞬时拥塞的情况，因为各种测量研究的结果表明[19, 27]，瞬时拥塞对于流媒体系统的设计来说是一个独特的挑战。读者可参阅参考文献［28］中对其他对等体动态模式的分析。

㊀ 这种假设是合理的，因为 P2P 流媒体系统中的对等体大致可以分为两级——高带宽的以太网对等体和低带宽的 DSL 对等体。虽然这里只假定了两级，但相应的分析可以很容易地引申到多级系统的情况。

在瞬时拥塞期间，在一个新的实时事件发布后的很短的时间内，将会有大量的对等体加入系统。为了模拟一个瞬时拥塞事件，我们假设时间被分为若干时隙，每个时隙播放一个分段。我们还假设所有的参与对等体在同一个时隙内加入该系统。需要强调的是，这些假设不是必须的，在分析中可以放宽条件。接下来介绍以下定义。

定义 3-1 瞬时拥塞等级的定义为：在一个瞬时拥塞事件发生期间加入系统的对等体的最大数量用 N 表示。

定义 3-2 令 δ 表示服务器强度，定义如下：

$$\delta = \frac{U_s}{NU_p}$$

其中，U_p是参与对等体的平均上传能力；U_s 是服务器上传能力。

正如 3.2.1 节中讨论的，R^2 不是一个严格的协议设计，因此在对 R^2 进行理论分析之前，应先列举出所有的设计选择。事实上，我们在分析中采用的是一个简单的随机 push 策略。更具体地说，当对等体有机会提供服务时，它会从它的邻居中均匀随机地选择一个伙伴，并上传这个伙伴最急需的分段中的一个编码块，这个分段指的是尚未完全接收的最接近播放点的分段。

下面的定理给出了在任意等级为 N 的瞬时拥塞期间流媒体能够以速率 R 流畅播放的充分条件。

定理 3-4[28] 假设基础覆盖网络完全连接。假定以下条件成立：

$$U_s + NU_p \geqslant (1+\varepsilon)NR \tag{3-3}$$

其中，ε 由下式给出：

$$\varepsilon = \gamma(q) + \frac{\ln(1+\delta) - \ln\delta}{k} \tag{3-4}$$

系统级参数 $\gamma(q)$表示由随机线性网络编码生成的线性相关的编码块的比例（取决于域的大小 q）。那么，当瞬时拥塞的等级 N 足够大时，在上述简单随机 push 策略下 R^2 能够以流速率 R 实现流畅播放。

将定理 3-4 与定理 3-2、3-3 结合，通过考虑可持续的流速率和初始缓冲延迟，可以更好地表征 R^2 与最优流媒体方案之间的性能差距。

推论 3-1 假设基础覆盖网络完全连接。那么在简单的随机 push 策略下，R^2 可以实现 $1+\varepsilon$ 倍的最大流率 R_{max}，其中 ε 由式（3-4）给出。

推论 3-2 假设基础覆盖网络完全连接。那么在简单的随机 push 策略下，R^2 可以实现 $2(1+\varepsilon)$倍的最短初始缓冲延迟，其中 ε 由式（3-4）给出。

推论 3-1 和推论 3-2 表明，即使采用简单的随机 push 策略，R^2 也能够在瞬时拥塞期间实现较短初始缓冲延迟并支持近优的速率。这里我们给出一个具体的数例来更好地说明这一点。正如参考文献［28］中说到的，当 q 较大时（$q \geqslant 64$），参数 $\gamma(q)$通常约为 0.1%。因此，在本例中取 $\gamma(q)$ = 0.1%。接下来，将服务器强度 δ 设为 0.001，将每个分段中的数据块数量 k 设为 100。则可持续的码流速率 R 满足 $R \geqslant R_{max}/1.07$，初始缓冲延迟在 2.14 倍限值的范围内。

注意，在上面的定理中，覆盖网络指的是一个完整的图。然而，在实际的流媒体系统中，每个参与对等体只能维持有限数量的邻居。因此，研究受限邻域的影响会是一个很有趣的问题。参考文献［28］中的仿真结果表明，小规模的邻域（如 50）已足够享有 R^2 的良好的整体性能。

3.2.4　基于网络编码的 P2P 多媒体流中的实际问题

R^2 的设计理念已经在悠视网得到了应用和实施。悠视网是一个由悠视网络公司（中国领先的对等互助媒体内容供应商之一）运营的大型操作流媒体系统。通过对大约 2000GB 的信息的实际收集和分析，参考文献［29］中的报道指出，在实践中使用网络编码可以实现理论上的优点：即允许多个上游对等体对一个下游对等体进行协作服务，这将带来缓冲延迟和带宽成本的最小化。特别指出的是，正常质量的视频在悠视网中的整体表现令人满意。对于高品质的视频，由于其较高的带宽需求，导致缓冲延迟可能较大。尽管如此，悠视网的测试表明，其延迟一般都在 20s 的合理范围内。

任何优势都会有随之而来的取舍问题。为了看清问题的全貌，我们现在来看看硬币的另外一面——R^2 的一些实际问题。与内容分发应用类似，网络编码的计算成本也是多媒体流应用必须重点关注的问题之一。即使使用现代处理器，它也不可能实现对包含超过几百个数据块的分段的计算。另一方面，如前所述，使用大段和小块是 R^2 成功的一个关键。因此，考虑到实际条件的制约，我们应在各段中尽可能多的增加数据块的数目，并最大限度地减少块的大小。

例如，R^2 中的每个分段被划分成 128 个数据块，每块 2KB。采用这种设计选择可以实现良好的综合性能，但代价是持续高的 CPU 利用率。换句话说，由于 CPU 的限制，我们不能在每个分段中划分更多数量的数据块。那么在 R^2 中可以选择较小的块长度吗？注意到在上述设计选择中，域大小 $q=256$ 时的“报头”的开销约为 6%。因此，当块长度更小时，可能会导致过多的开销。总之，在性能、计算成本和开销之间存在一个折衷，我们需要针对特定的流媒体系统寻找最合适的折衷关系。

R^2 的另一个实际问题是制动冗余。回想一下，一旦下游对等体接收完一个完整的分段，它会立即给所有的邻居发送一个新的缓存映像，该缓存映像也是一个使上游对等体停止此段服务的信号。由于制动信号到达上游对等体需要一定的时间，因此下游对等体在接收到完整的分段后可能还会收到额外的冗余块。我们应如何尽量减少这种制动冗余呢？一种工程方法是采用“提前制动”机制，鼓励下游对等体在没有完全接收完分段之前就通知其所有上游对等体的一个子集停止服务。然而，对这样一个“提前制动”算法的设计仍然是一个挑战。

最后，R^2 还可能有一个缺点。由于同步播放特性，实时事件的发生和播放之间的时间对于所有的参与对等体都是一样的。虽然这会对一些应用程序产生危害，但这对于涉及实时互动的应用（如现场投票与 SMS）可能是一个优势：所有的参与对等体会同时看到相同的内容，因此相应的互动行为也可以同时开始产生。

3.3 总结

本章的主要目的是探索网络编码在 P2P 网络中可能提供的潜在效益。我们重点讨论了两个主要应用：内容分发和多媒体流。为了实现这一目标，我们首先介绍了如何将网络编码成功地运用到每个应用中。接着提出了一些应用网络编码的重要优势。

- 对于 P2P 内容分发，使用网络编码能够简单而有效地解决数据块调度问题，并实现了较短的文件下载时间和对于对等体离开的更好的鲁棒性。
- 对于 P2P 多媒体流，需要完全重新设计流媒体协议才能充分利用网络编码。我们特别提出了一个新的基于网络编码的流媒体系统设计——R^2，并解释了为什么 R^2 能够使用网络编码来充分利用现有的带宽资源，从而提高系统的整体性能。

为了加深对上述优势的理解，我们选择性地给出了一些理论成果。最后，提出了若干在现实世界系统设计中值得特别注意的实际问题。我们相信，这样的探索能够揭示未来网络编码在 P2P 网络中的应用。

参 考 文 献

[1] B. Cohen. Incentives Build Robustness in BitTorrent. Proc. of 1st Workshop on Economics of Peer-to-Peer Systems, (Berkeley, CA), June 5 - 6, 2003.

[2] J. Pouwelse, P. Garbacki, D. Epema, and H. Sips. The Bittorrent P2P File - sharing System: Measurements and Analysis. Proc. of 4th International Workshop on Peer - to - Peer Systems (IPTPS), (Ithaca, New York), Feb. 24 - 25, 2005.

[3] C. Gkantsidis and P. Rodriguez. Network Coding for Large Scale Content Distribution. Proc. of IEEE INFOCOM 2005, (Miami, FL), March 13 - 17, 2005.

[4] C. Gkantsidis, J. Miller, and P. Rodriguez. Anatomy of a P2P Content Distribution System with Network Coding. Proc. of 5th International Workshop on Peer - to - Peer Systems (IPTPS), (Santa Barbara, CA), Feb. 27 - 28, 2006.

[5] R. W. Yeung. Avalanche: A network coding analysis. Communications in Information and Systems, vol. 7, pp. 353 - 358, 2007.

[6] Yunnan Wu. A Trellis Connectivity Analysis of Random Linear Network Coding with Buffering. Proc. of International Symposium on Information Theory (ISIT), (Nice, France), June 24 - 29, 2007.

[7] T. Ho, R. Koetter, M. M'edard, M. Effros, J. Shi, and D. Karger. Arandom linear network coding approach to multicast. IEEE Trans. Inf. Theory, vol. 52, no. 10, pp. 4413 - 4430, October 2006.

[8] A. G. Dimakis, P. B. Godfrey, Y. Wu, M. Wainwright, and K. Ramchandran. Network coding for distributed storage systems. IEEE Trans. Inf. Theory, vol. 56, no. 9, pp. 4539 - 4551, Sep. 2010.

[9] S. Acedanski, S. Deb, M. M'edard, and R. Koetter. How Good is Random Linear Coding Based Distributed Networked Storage? First Workshop on Network Coding, Theory, and Applications (NetCod), (Riva del Garda, Italy), Apr. 2005.

[10] S. Deb, M. M'edard, and C. Choute. Algebraic gossip: A network coding approach to optimal multiple rumor mongering. IEEE Trans. Inf. Theory, vol. 52, no. 6, pp. 2486 – 2507, June 2006.

[11] YunnanWu, Y. C. Hu, J. Li, and P. A. Chou. The Delay Region for P2P File Transfer. Proc. of International Symposium on Information Theory (ISIT), (Coex, Seoul, Korea), June 28 – July 3, 2009.

[12] C. S. Chang, T. Ho, M. Effros, M. M'edard, and B. Leong. Issuesin Peer – to – Peer Networking: a Coding Optimization Approach. Proc. of the 2010 IEEE International Symposium on Network Coding (Net – Cod), (Toronto, Canada), June 9 – 11, 2010.

[13] P. Chou, Y. Wu, and K. Jain. Practical Network Coding. Proc. of Allerton Conference on Communication, Control, and Computing, (Monticello, Illinois), Oct. 1 – 3, 2003.

[14] P. Maymounkov, N. J. A. Harvey, and D. S. Lun. Methods for Efficient Network Coding. Proc. of Allerton Conference on Communication, Control, and Computing, (Monticello, Illinois), Oct. 1 – 3, 2006.

[15] D. Niu and B. Li. On the Resilience – Complexity Tradeoff of Network Coding in Dynamic P2P Networks. Proc. of 15th IEEE International Workshop on Quality of Service (IWQoS), (Evanston, IL), June 21 – 22, 2007.

[16] M. Wang and B. Li. How Practical is Network Coding? . Proc. of 14th IEEE International Workshop on Quality of Service (IWQoS), (New Haven, CT), June 19 – 21, 2006.

[17] R. Koetter and F. R. Kschischang. Coding for errors and erasures in random network coding. IEEE Trans. Inf. Theory, vol. 54, no. 8, pp. 3579 – 3591, Aug. 2008.

[18] V. Venkataraman, K. Yoshida, and P. Francis. Chunkyspread: Heterogeneous Unstructured Tree – based Peer – to – Peer Multicast. Proc. of 14th IEEE International Conference on Network Protocols (ICNP), (Santa Barbara, CA), Nov. 12 – 15, 2006.

[19] S. Xie, B. Li, G. – Y. Keung, and X. Zhang. Coolstreaming: Design, theory, and practice. IEEE Transactions on Multimedia, vol. 9, pp. 1661 – 1671, December 2007.

[20] Y. Huang, Z. J. Fu, D. M. Chiu, C. S. Lui, and C. Huang. Challenges, Design and Analysis of a Large – scale P2P VoD System. Proc. of ACM Sigcomm, (Seattle, WA), Aug. 17 – 22, 2008.

[21] M. Wang and B. Li. Lava: A Reality Check of Network Coding in Peer – to – Peer Live Streaming. Proc. of IEEE INFOCOM, (Anchorage, Alaska), May 6 – 12, 2007.

[22] M. Wang and B. Li. R2: Random push with random network coding in live peerto – peer streaming. IEEE J. Sel. Areas Comm. , vol. 25, no. 9, pp. 1655 – 1666, Dec. 2007.

[23] M. Zhang, Q. Zhang, L. Sun, and S. Yang. Understanding the power of pull – based streaming protocol: Can we do better? IEEE J. on Sel. Areas in Communications, vol. 25, pp. 1678 – 1694, Dec. 2007.

[24] C. Feng, B. Li, and B. Li. Understanding the Performance Gap between Pull – based Mesh Streaming Protocols and Fundamental Limits. Proc. of IEEE INFOCOM, (Rio de Janeiro, Brazil), April 19 – 25, 2009.

[25] L. Guo, S. Chen, Z. Xiao, E. Tan, X. Ding, and X. Zhang. Measurements, Analysis, and Modeling of Bit Torrent – like Systems. Proc. of Internet Measurement Conference (IMC), (Berkeley, CA),

Oct. 19 – 21, 2005.

[26] R. Kumar, Y. Liu, and K. W. Ross. Stochastic Fluid Theory for P2P Streaming Systems. Proc. of IEEE INFOCOM, (Anchorage, Alaska), May 6 – 12, 2007.

[27] X. Hei, C. Liang, J. Liang, Y. Liu, and K. W. Ross. A measurement study of a largescale P2P IPTV system. IEEE Trans. Multimedia, vol. 9, no. 8, pp. 1672 – 1687, Dec. 2007.

[28] C. Feng and B. Li. On Large – Scale Peer – to – Peer Streaming Systems with Network Coding. Proc. of ACM Multimedia, (Vancouver, BC), Oct. 27 – Nov. 1, 2008.

[29] Zimu Liu, Chuan Wu, Baochun Li, and Shuqiao Zhao. UUSee: Large – Scale Operational On – Demand Streaming with Random Network Coding. Proc. of IEEE INFOCOM, (San Diego, California), March 15 – 19, 2010.

第 4 章　真实世界中的网络编码

Janus Heide　丹麦，奥尔堡，奥尔堡大学，电子系统系
Morten V. Pedersen　丹麦，奥尔堡，奥尔堡大学，电子系统系
Frank H. P. Fitzek　丹麦，奥尔堡，奥尔堡大学，电子系统系
Torben Larsen　丹麦，奥尔堡，奥尔堡大学，电子系统系

摘要

本章将讨论网络编码在商业移动平台中的实现，并重点讨论在移动电话上的实现。由于移动电话受限于存储空间、能量供给以及计算能力，因此在移动电话上实现网络编码面临着诸多挑战。这些挑战开启了新的研究方向，引出了许多理论和实践上的研究问题。

关键词

关键词：蜂窝网络、Ad Hoc 网络、内容分发网络、性能评估、实现问题、最优化、码型设计。

4.1　引言：并非火箭科学

任何时候人们想要强调某件事情容易时，总引用同样的话语："并非火箭科学"。那火箭科学最大的难题是什么？如果熟悉能量守恒和牛顿定律等物理基本定理的话，就很容易推导出将火箭发射到太空的基本公式。但问题并不在于理论工作而在于现实世界中的具体实现。即使你清楚火箭的重量以及所需燃料的数量，如何在火箭中正确使用燃料也仍然是一个问题。因此如何在现实世界中实现理论知识是一个非常重要的研究领域，尤其是当现实实现能够为理论工作提供有价值的反馈的时候。

本章将关注网络编码在移动网络中的应用。蜂窝电话因其广泛适应性、高可用性和高可移动性等特点成为一个令人兴奋的平台。但是与服务器或桌面系统相比，移动终端的计算能力有限，此外，移动电话有限的能量供给也使得在设计过程中能量消耗成为主要的考量点。具体地说，所有通过使用网络编码得到的吞吐量增益都应在增加能量消耗方面付出最小的代价。然而，目前针对许多功能更强大的平台所提出的方法都无法直接应用到移动电话网络中，因此我们需要针对移动设备的特点研究新的技术。

4.2　移动电话上的网络编码

为了推动对网络编码应用的研究，我们首先给出一些场景，这些场景可以表现出网络编码技术在移动电话网络中的潜在应用和优势。

移动电话可以创建、存储和交换多媒体内容。目前，许多移动电话都可以把内容上传到云服务，如 Facebook、Twitter、MobileMe 等。一般情况下，内容生成者首先会上传内容，然后把内容链接发送给其他用户。但是这可能导致较长的延迟，并加剧上行和下行通道的拥塞，这种情况同样也发生在核心网络。

当一群人比邻而居时，他们可能想要消费相同的内容，如在参考文献［1］中提出的彩色移动应用程序。这些应用可能希望在没有任何覆盖网络的情况下能够在移动电话之间直接分发内容。内容的直接分发可以降低传输延时，同时也可以减少基站和核心网络的流量。对于分布式照片、音乐、视频、文件和其他类型的内容采用直接内容分发更为有效。参考文献［2，3］中给出了一个系统，系统中的移动终端通过一跳或多跳移动网络将数据直接发送给其邻近设备。从一台设备到多台接收机的视频广播示意图如图4-1所示。参考文献［2，3］主张采用网络编码在无线设备之间直接分发内容，并表明网络编码技术是有效实现这类系统的工具之一。



图4-1 从一台设备到多台接收机的视频广播

在单跳网络中，我们可以通过单播或广播连接实现内容的交换。对于单播连接，带宽和能量的消耗取决于接收设备的数量。从能量和带宽的角度考虑，特别是对于大用户群，由于多台接收机可以同时从单次传输中接收数据，故数据广播是更有效的解决方案。对于容易出错的通信链路，需要通过一定形式的编码来恢复数据的错误和消失的数据。

此外，如果接收节点之间可以协同转发数据，则服务器的负荷就会显著降低。当内容源距离较远且通信链路容易发生错误时，协同下载就会非常有利（见图4-2）。这种情形下，网络编码技术可以用于促进协同设备之间的数据分发。实际上，使用标准的编

码技术时，服务器需要发送多余的数据才能使节点具有错误恢复的功能，而使用网络编码技术时，每台接收设备都可以转发再编码[⊖]后的信息给其他节点，减轻服务器的负担。

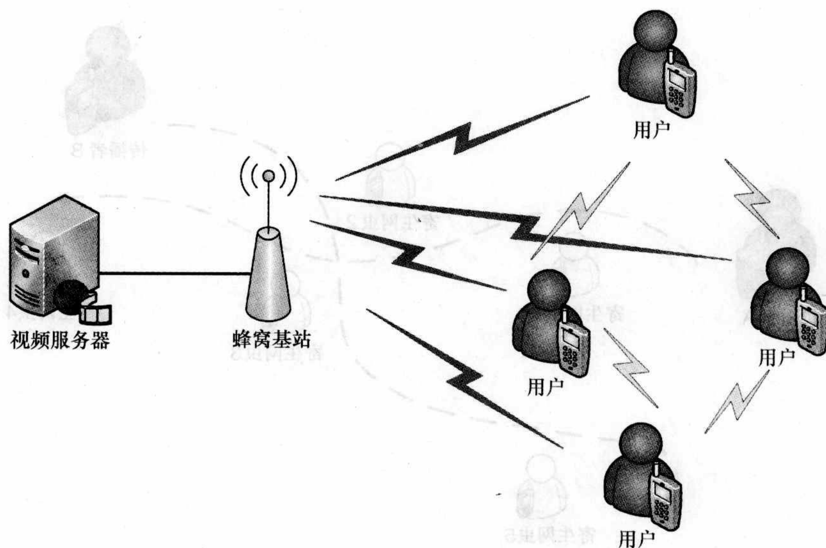


图 4-2　多跳网络中的内容分发

在一些发展中国家，其蜂窝覆盖有时是稀疏或不可靠的，此时可以利用网络编码技术来实现内容分发。假设有一用户（称其为传播者），他/她在旅行时想要登录城市热点并将其设备中的内容以信息传输的形式传播给其他用户，这些用户没有直接连接到网络，被称为寄生网虫。通过使用网络编码技术，寄生网虫能有效地从不同的传播者那里获得内容。网络编码允许寄生网虫从不同的源获得内容而不会出现“赠券收集问题”。每次从整套赠券中抽取一张（具有替换者），目的是集齐该套中所有不同的赠券的问题，被称为“赠券收集问题”。起初，收集到独特的（之前没有的）赠券的几率很大，随着已收集的赠券越来越多，再收集到独特赠券的可能性就会越来越低。

图 4-3 给出了另一个实例，两个传播者 A 和 B 要通过多跳路由将信息传送给不同的寄生网虫。寄生网虫 1、4 和 5 只接收来自单个传播者的数据包，寄生网虫 2 和 3 则从两个传播者处获得他们的数据包。本例中，网络编码可以帮助减少传播者和寄生网虫之间的信号传输总量。假设寄生网虫 2 和 3 将来自传播者 B 和寄生网虫 1 的数据包分别转发给寄生网虫 4 和 5。使用传统方法时，为了保证发送的信息对于接收设备是有用的，寄生网虫 2 和 3 需要相互协调。如果采用网络编码技术，节点们仅需简单地组合它

⊖ 设备再编码指的是它对自身拥有的部分信息进行组合，其方式与信源在编码时组合信息的方式相似，我们将在第 4.3 节详细介绍关于再编码的内容。

们已经收到的所有信息，然后把组合作为新的数据包发送出去，这样几乎就不需要协调。正因为这种组合包含了许多数据包，因此接收端从接收到的数据包中能够提取出新信息的概率就大增。

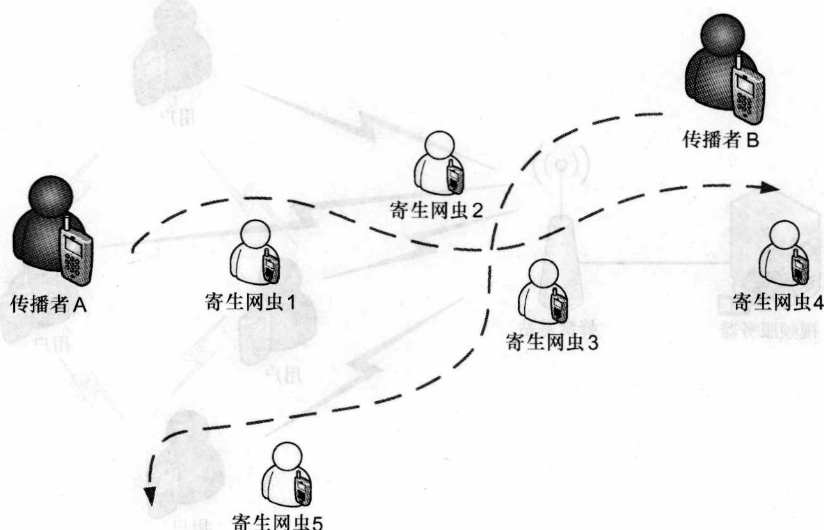


图 4-3　病毒式的内容传播

网络编码的另一应用领域是分布式存储。我们先前的工作[4]已经表明，通过将内容分布到多个移动设备可以改善传感器网络的可靠性。这种方法的好处是，即使一些设备丧失功能或电能耗尽，仍然会有足够的设备可以发送信息。图 4-4 中所示的是一个移动存储网络示例。系统中有一个聚合节点，它从不同的传感器上接收数据并将部分信息存储在不同的移动设备上。为了降低对远端设备存储空间的需求，每个设备上只存储一部分信息。如果不采用网络编码，我们就必须设计复杂的信号协议来保证系统对抗磁盘失效的健壮性。例如，假设我们需要存储 100 个不同的信息数据包，采用网络编码技术时将生成 300 个编码数据包并把它们分布存储到所有的节点中。当一些节点丢失时，只要能收集到 100 个不同的编码数据包（这种可能性还是比较高的），我们就仍然能解码出所有的存储信息。

参考文献［5，6］指出网络编码可应用于多路径接收，其主要思想是通过增加额外的空中接口，移动设备可以动态地增加信道容量。用 WiFi 接口接收视频作为实例，当 WiFi 链路容易发生错误或视频本身需要额外带宽时，移动设备可用蜂窝链路来传输额外的数据，如图 4-5 所示。当采用传统方法时，我们需要通过协调机制来保证性能，并确保无线 WiFi 和蜂窝链路传输的是不同的数据包；而网络编码技术可以通过在内容服务器端混合数据解决这一问题。

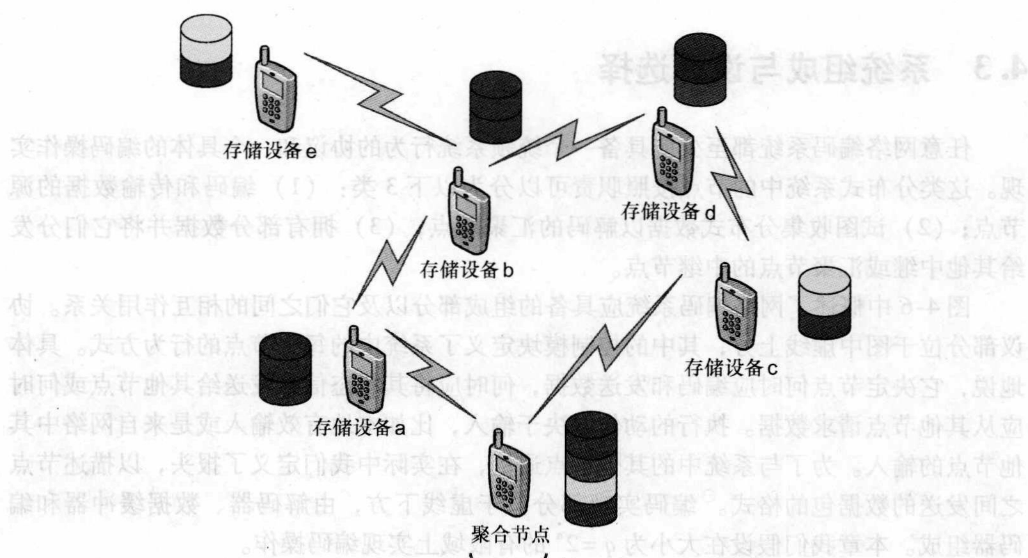


图 4-4　移动分布存储

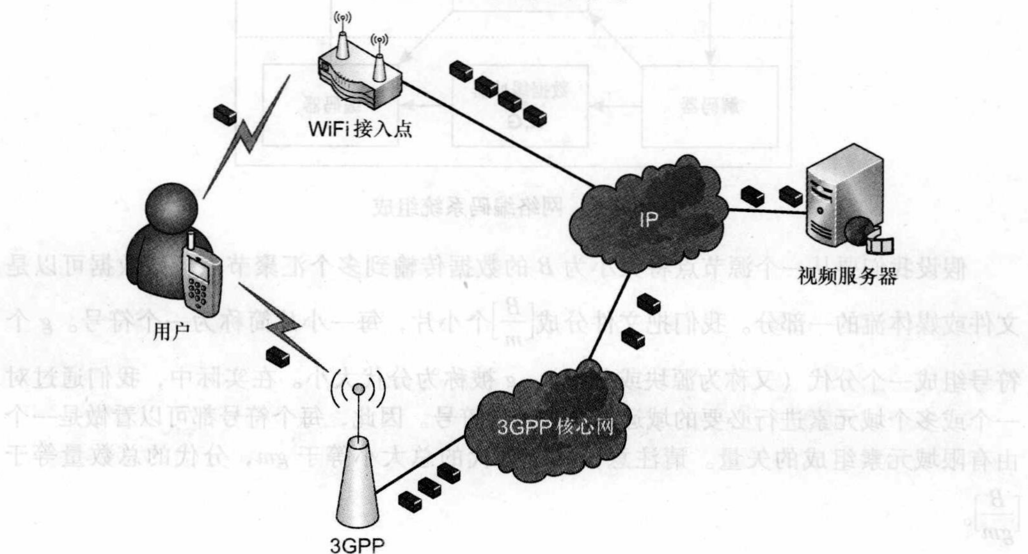


图 4-5　多路径接收

4.3 系统组成与设计选择

任意网络编码系统都至少应具备一个统领系统行为的协议和一个具体的编码操作实现。这类分布式系统中的节点按照职责可以分为以下 3 类：（1）编码和传输数据的源节点；（2）试图收集分布式数据以解码的汇聚节点；（3）拥有部分数据并将它们分发给其他中继或汇聚节点的中继节点。

图 4-6 中概述了网络编码系统应具备的组成部分以及它们之间的相互作用关系。协议部分位于图中虚线上方，其中的控制模块定义了系统中的每个节点的行为方式。具体地说，它决定节点何时应编码和发送数据，何时应将其状态信息发送给其他节点或何时应从其他节点请求数据。执行的动作取决于输入，比如本地有效输入或是来自网络中其他节点的输入。为了与系统中的其他节点通信，在实际中我们定义了报头，以描述节点之间发送的数据包的格式。[⊖]编码实现部分位于虚线下方，由解码器、数据缓冲器和编码器组成。本章我们假设在大小为 $q=2^n$ 的有限域上实现编码操作。

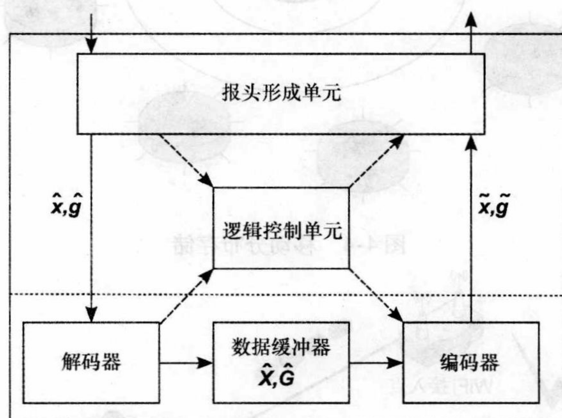


图 4-6 网络编码系统组成

假设我们要从一个源节点将大小为 B 的数据传输到多个汇聚节点。该数据可以是文件或媒体流的一部分。我们把文件分成$\left[\frac{B}{m}\right]$个小片，每一小片简称为一个符号。g 个符号组成一个分代（又称为源块或分批），g 被称为分代大小。在实际中，我们通过对一个或多个域元素进行必要的域运算来表示各符号。因此，每个符号都可以看做是一个由有限域元素组成的矢量。请注意，一个分代的总大小等于 gm，分代的总数量等于$\left[\frac{B}{gm}\right]$。

⊖ 本章节将不描述具体的格式。

将一个分代中的 g 个长度为 m 的原始符号组成矩阵 $\boldsymbol{M}=[\boldsymbol{m}_1;\ \boldsymbol{m}_2;\ \cdots;\ \boldsymbol{m}_g]$，其中 $\boldsymbol{m}_i$ 表示该分代中的第 i 个符号。最前面的 g 个符号或 gm 个字节的数据组成了标号为 **0** 的分代。

要对某一分代生成一个新的编码符号 $\boldsymbol{x}$，源节点需要把矩阵 $\boldsymbol{M}$ 和一个长度为 g 的编码矢量 $\boldsymbol{g}$ 相乘，即 $\boldsymbol{x}=\boldsymbol{M}\boldsymbol{g}$。我们可以通过这种方式构造出 $g+r$ 个编码符号（每个编码符号都和一个编码矢量相关联），其中 r 表示冗余符号的数量。编码符号和其对应的编码矢量将被一起组成编码包，并在网络中传输。由于一些实现方面的原因，每个编码符号都是某一分代中的原始符号的一个（线性）组合。这样做的好处是可产生的编码符号的数目不受限。由于该码的速率不固定，故也称之为无速率码。在随机线性网络编码（RLNC）中，编码矢量的产生是随机的，而在一些确定编码方法中，编码矢量是基于一些预定义的算法产生的。因此，要实现编码，我们需要随机数产生器、加法器和乘法器㊀。

汇聚节点若要成功解码某个分代，它必须接收到 g 个该分代的线性独立的符号和编码矢量。所有接收到的符号可以组成矩阵 $\hat{\boldsymbol{X}}=[\hat{\boldsymbol{x}}_1;\ \hat{\boldsymbol{x}}_2;\ \cdots;\ \hat{\boldsymbol{x}}_g]$，所有的编码矢量可以组成矩阵 $\hat{\boldsymbol{G}}=[\hat{\boldsymbol{g}}_1;\ \hat{\boldsymbol{g}}_2;\ \cdots;\ \hat{\boldsymbol{g}}_g]$，称为解码矩阵。原始数据 $\boldsymbol{M}$ 可以被解码为 $\hat{\boldsymbol{M}}=\hat{\boldsymbol{X}}\hat{\boldsymbol{G}}^{-1}$。通常，如果接收到同一分块的编码符号的个数达到 g 个或更多，则可解码出该分组的原始数据。不使用编码时，接收端必须收集全部 g 个不同的原始符号，相比而言，使用网络编码的接收条件要宽松许多。

任一接收到同一分代中的 $g'\geqslant 2$ 个线性独立符号的节点均可生成一个新的数据包（这一操作称为再编码）。所有接收到的符号组成矩阵 $\hat{\boldsymbol{X}}=[\hat{\boldsymbol{x}}_1;\ \hat{\boldsymbol{x}}_2;\ \cdots;\ \hat{\boldsymbol{x}}_g{}']$，所有的编码矢量组成矩阵 $\hat{\boldsymbol{G}}=[\hat{\boldsymbol{g}}_1;\ \hat{\boldsymbol{g}}_2;\ \cdots;\ \hat{\boldsymbol{g}}_g{}']$。为了产生新的符号 $\tilde{\boldsymbol{x}}$，需要将矩阵 $\hat{\boldsymbol{X}}$ 和一个随机产生的长度为 g' 的矢量 $\boldsymbol{h}$ 相乘（即 $\tilde{\boldsymbol{x}}=\hat{\boldsymbol{X}}\boldsymbol{h}$），其相应的编码矢量等于 $\tilde{\boldsymbol{g}}=\hat{\boldsymbol{G}}\boldsymbol{h}$。我们可以通过这种方式构造出 r' 个随机产生的再编码符号（每个符号都和一个编码矢量相关联），节点可以产生 $r'>g'$ 个符号，但其中只有 g' 个是相互独立的。注意 $\boldsymbol{h}$ 仅在本地使用，没有必要区分编码符号和再编码符号。在实际操作中，这意味着接收到多于一个符号的节点可以把这些符号重新组合成再编码符号，如同在源节点上构造编码符号一样。

图4-7给出了一个网络编码系统以及编码符号在信道上消失后的具体操作。编码器利用 g 个原始源符号生成 k 个编码符号。编码符号通过不可靠信道传输，该信道总是丢失一定数量的符号，接收端收集足够其解码所需数量（一般来说是略多于 g 个）的符号后解码出原始数据。网络编码方案和喷泉编码的显著不同是前者可以在中间节点（发送端和接收端之间）上编码。在再编码过程中，中间节点将重新组合它接收到的不完整的数据并生成再编码符号，图底部的右角处给出了该过程的示意。

㊀ 所有的操作均定义在有限域上。

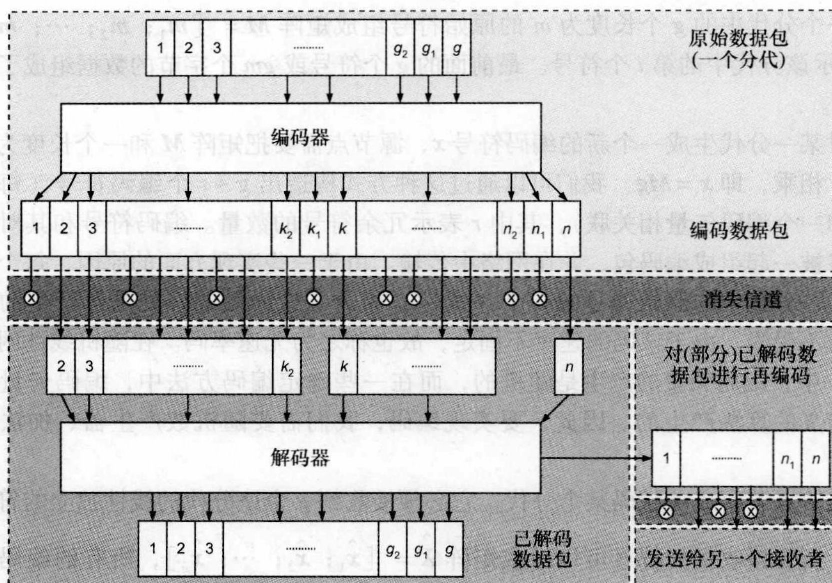


图 4-7 存在丢码的网络编码系统

4.4 实际问题

大多数操作系统都可以提供随机数生成器，但不幸的是这些产生的数据并不总是充分随机（见图 4-8）。这会导致较高概率的线性相关，进而影响系统的性能。避免这类问题的最简单的办法就是采用参考文献［7］中给出的被证实性能可接受的随机数发生器，例如参考文献［8］中给出的对 MersenneTwister 的实现。



图 4-8 呆伯特的随机性

在二进制扩展域上，加法和减法运算等价于比特位的异或操作，我们可以有效地执行它们。而乘法和除法运算则较为复杂。一种在高阶域上实现的方法是采用查找表，这种情形下，乘法和除法运算可以通过一系列查找表和一些其他的基本操作加以实现。参考文献［9］给出了若干种不同的查找表方法和少量编码实例。为了让这些方法能够获得良好的性能，这些查找表应保存在 CPU 的快速缓冲器中，否则频繁地存取内存将会

降低编码吞吐量。为此，我们需要尽量减小表的大小，而这又取决于相应的有限域的大小。

此外，还可以通过将两个多项式（代表两个有限域上的符号）相乘，再进行以一个不可约多项式为底的模运算来实现乘法和除法运算。除法也可以利用欧几里德算法实现（细节和源码实例可参阅参考文献［10，p. 14，122］）。这些算法可以通过一些简单的操作来得到计算结果。这两种方法的性能都依赖于实现它们的平台，特别是中央处理器和内存存取的相对速度，我们还可以采用一些额外的优化技术。

矩阵的逆运算实现可以有多种不同的算法，最常用的一种是 Gauss - Jordan 算法。该算法简单且可用于对部分数据解码，由于接收端每时隙只能收到一个数据包，因此这一优点能够降低最终的解码延迟。还有其他算法可参见参考文献［11］中的示例。但是，这些方法往往要求求逆的矩阵具有一些特定的属性，或是只在某些情况下才具有更好的性能。

为了比较不同的编码方法和优化技术，我们查看了现有的关于网络编码的编码和解码操作的文献。由于解码所需的运算量要远远大于编码，因此我们重点考虑解码吞吐量。为了对该领域的最新成果有一个总览，我们将粗略地按照时间顺序给出一些研究成果。

4.5 二进制确定性方法

网络编码的第一个移动实现是基于确定性/机会性二进制网络编码[12]。在这类系统中，网络编码用于改善传统广播网络的性能。当系统检测到对相对较少的数据包一起进行异或运算具有编码潜力时，系统就会进行编码和再编码。如果检测到通过编码可以提高系统的吞吐量，即意味着存在编码潜力。编码数据包可以在汇聚节点依次被解码。因为我们只通过异或操作对少量数据包进行编码，故其编、解码操作不需太多计算量。这是很有优势的，尤其是对移动电话来说。当实际中的网络是分布式系统时，网络中的节点只具有不完全的系统状态信息，故而很难检测到编码机会。因此，在这样一个系统中编码的实现相对简单但逻辑控制十分复杂。

参考文献［13］中介绍的 COPE 架构采用了确定性/机会性二进制流间网络编码。网络编码技术被应用于数据链路层和网络层之间。系统是针对由笔记本电脑组成的无线 mesh 网络设计的。这一架构无法直接应用到移动电话上，因为移动电话的网络栈大体上是封闭的，对网络栈的任何改动均需要手机厂商的协作。此外，COPE 架构很大程度上依赖于对数据包的监听，因此，每个汇聚节点需要接收的数据包的数量都较大，从而导致能量消耗的增加。这也表明了针对 PC 和移动电话设计的系统有不同的性能标准。

基于 COPE 的启发，参考文献［12］中提出了一个系统，该系统适用于移动电话领域和另一种分布式场景。研究人员在诺基亚 N95 手机上实现并测试了该系统。该网络编码系统在应用层采用了确定性/机会性二进制流间网络编码，以改进从单源到多个协作汇聚节点的内容分发。一个源节点通过全局无线链路将相同的内容发送给一组汇聚节点，为了使汇聚节点间可以协作，源节点把其内容分割成不同的流，这种技术通常称为

内容分割。汇聚节点之间可以通过本地无线连接交换数据形成协作。为了提高本地交换的效率，汇聚节点一检测到编码机会就会进行再编码。

4.6 随机线性网络编码

随机线性网络编码（RLNC）的构造基于较大的有限域上的密集随机编码矢量。使用随机网络编码时，每个编码数据包都是由给定有限域上的若干数据包组合而成的，典型的有限域大小为2^8、2^{16}和2^{32}。在这类有限域上实现快速操作是一个非凡的任务。因此，实现编码和解码高吞吐量是我们面临的主要挑战，特别是在计算能力较低的平台上。参考文献［14－16］中提出了首个针对计算机的实现。

参考文献［17］中给出了在采用主频为332MHz的ARM11中央处理器的诺基亚移动电话N95上的基本实现和测试应用。该实现是基于对数和指数函数查找表，并通过改进的Gauss－Jordan算法实现矩阵求逆。使用的域的大小为2^8和2^{16}，测试的分代大小在10~400之间，数据被编码为大小等于1.5KB的数据块，这样无需分割就可以通过无线局域网进行传输。

测试结果如图4-9所示。当分代大小同为10时，有限域大小采用2^{16}，比采用2^8时的编码吞吐量大约高出50%。当分代大小较大时，采用大小为2^{16}的有限域带来的吞吐量增益非常有限。当分代大小大于64时，解码吞吐量低于20KB/s。因此，该结果表明上述实现的性能不是很乐观。

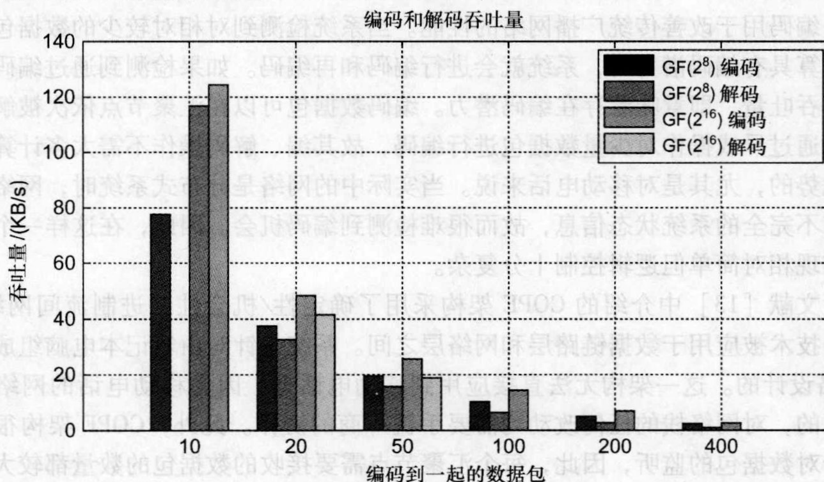


图4-9　在诺基亚N95移动电话上基于查找表实现的编码吞吐量。分代大小在10~400间，数据块大小为1.5KB

参考文献［18］中给出了在iPhone3G和二代iPodTouch上的系统实施和部署，iPhone3G和二代iPodTouch分别采用主频为412MHz和533MHz的ARM11中央处理器，二者均基于对数和指数的查找表实现。此外，我们还提出了一种基于环的方法，该方法

中的乘法运算可通过一系列基本运算实现。对于 CPU 的高速缓存中存有查找表的情况[⊖]，这可能是一个优势。这种情况下可能会出现大量的高速缓存缺失，这将显著降低基于查找表的方法的性能。矩阵逆运算通过 Gauss - Jordan 算法实现。在测试中我们令域的大小为 2^8，分代大小为 64 ~256，编码数据块的大小则介于 128B ~16KB 之间。此外，研究表明利用单指令多数据（SIMD）指令对该实现进行优化可以显著改善吞吐量。

该系统中，当分代大小为 64 时，解码吞吐量可达 370KB/s。基于循环方法实现的吞吐量大约是查找表方法的两倍。作者对测试进行了不同的调整，以获得以上结果，例如手动优化编译程序。作者还对一个速率为 77. 5KB/s 的多媒体流进行了解码和播放实验，以观察能量消耗，实验结果表明网络编码会增加约 50% 的能量消耗。

以上观测到的编码吞吐量是否可以接受取决于具体的应用。首先值得注意的是，这些吞吐量值是在 CPU 占用率为 100% 的情况下得到的，而大多数应用会利用 CPU 周期来工作而不是传输数据。其次，高 CPU 使用率将导致高能耗。因此，我们需要进一步优化网络编码技术，使其适合于范围更广泛的应用。

4.7　通过优化提升随机线性网络编码

我们可以通过不同的优化技术来提高编码吞吐量。例如，现代 CPU 使用的 SIMD 指令集，多核 CPU 使用的并行处理技术以及支持网络编码的专用硬件。SIMD 指令集可有效应用于当前的大多 CPU 上，包括移动电话使用的 CPU。相比于串行处理，基于多核的并行处理可以依照其有效内核的数目成倍地提高吞吐量。目前还没有集成了多核处理器的移动电话，因此并行处理不会带来帮助。但是，在不久的将来如果 PC 上的多核趋势蔓延到移动电话领域，这一情形将大为改观。专用硬件尤其值得关注，因为它可以在更少功率消耗的条件下允许极其快速的编码，但其额外的硬件成本让人望而却步。在个人计算机领域已经提出的一个解决方案是使用图形处理单元（GPU）。使用该方案时，现有的硬件可被重复利用来完成新的任务，这是一种实现网络编码的有趣而廉价的方式。

到目前为止，只有单指令多数据指令（SIMD）被应用于移动电话领域[18]，而多线程处理和 GPU 的使用实验只是在 PC 上展开，但在不久的将来这些技术可能会被应用到手机上。本章中我们仅研究计算机实现。目前只有少数手机有多核处理器，但有许多手机都集成了图形处理器（GPU）。

参考文献［19］中给出了在如下硬件平台上的并行处理实现：4 核 CPU IntelPentium 4 Xeon 2. 8GHz；双核 CPU Intel Pentium 4 Xeon 3. 6GHz；双核 CPU PowerPC G5 2. 5GHz。对于这些平台分别给出了采用基于对数和指数的查找表以及基于环的实现方法。矩阵逆运算通过改进的 Gauss - Jordan 算法实现。采用 SIMD 指令集优化后的基于环的方法显著优于基于查找表的方法。为了利用平台的多核架构，并行处理得以实现。域的大小为 2^8，

⊖　随着域大小的增加，查找表的大小也随之增加。另外，一般来说相比于个人计算机来说，移动电话的 CPU 的高速缓冲器明显较小。

分代大小介于 64 ~ 256 之间，编码数据块大小介于 128B ~ 32KB 之间。当分代大小为 64 时，解码吞吐量可达 43MB/s。与单线程方法比较，采用并行方法㊀在解码吞吐量方面获得的改变是：当数据块大小取最小值时，吞吐量略有下降，但当数据块大小取最大值时，吞吐量获得了约 5 倍的提升。

参考文献［20］给出了一个在 Nvidia CUDA 上的实现，测试配置包括一台使用 Intel Q6600 2.4GHz 4 核 CPU 的 PC 和一个包含 192 个以 1.92GHz 频率运行的核的 Nvidia 260 GTX 显卡。该实现同时利用 CPU 和 GPU 来解码数据包。单线程和多线程矩阵求逆技术在 CPU 上实现，解码数据所需的乘法操作则由 GPU 完成。将 CPU 和 GPU 结合用于解码主要是为了克服 GPU 中的线程同步问题。该实现采用基于对数和指数查找表的方法，矩阵逆运算通过 Gauss - Jordan 算法实现㊁，域大小是 2^8，分代大小介于 128 ~ 512 之间，编码数据块大小介于 1 ~ 32KB 之间。

参考文献［20］的结果如图 4-10 所示㊂。其中当分代大小为 128，数据块大小为 32KB 时，解码吞吐量高达 225MB/s。随着分代大小的增加和数据块大小的减少，编码吞吐量明显下降。观察表明，CPU 上的矩阵求逆是非常主要的性能瓶颈。

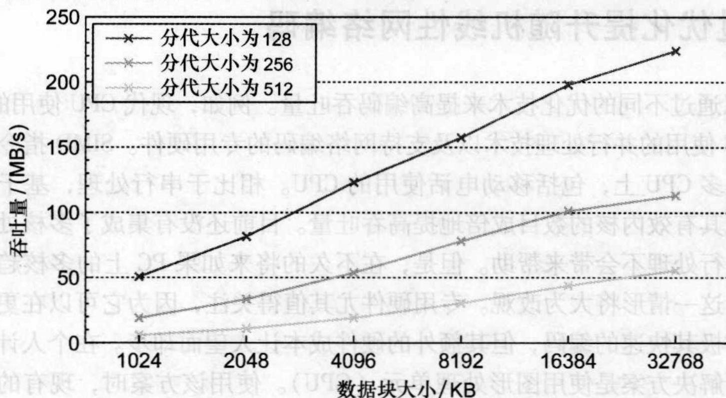


图 4-10　一个配备了 Nvidia 260 GTX 显卡的强大的个人 PC 上的解码吞吐量。测试的分代大小为 128、256、512，数据块大小介于 1 ~ 32KB 之间[20]

参考文献［21］同样给出了一个在 Nvidia CUDA 上的实现，其测试配置是一个 Dual Intel Xeon 2.8GHz 四核 CPU 和一个包含 112 个以 1.5GHz 频率运行的核的 Nvidia 8800 GT 显卡。该实现使用的是基于环的方法，同时测试了只使用 GPU 解码和使用 CPU 辅助 GPU 解码的情形。CPU 辅助 GPU 解码的性能要好于纯 GPU 方案。作者还讨论了与纯 GPU 编码相关的一些问题，并与参考文献［19］中给出的基于 CPU 的实现进行了比较。矩阵逆运算通过改进的 Gauss - Jordan 算法实现，域大小是 2^8，分代大小介于 128 ~ 512 之间，编码数据块大小介于 128B ~ 16KB 之间。

㊀ 二者均基于相同的用 SIMD 优化的基于环的方法。

㊁ 求逆矩阵的方法还没有直接在出版物中叙述，但已被作者证实。

㊂ 感谢 Xiaowen Chu 提供此数据。

在参考文献［22］中，作者通过在更强大的拥有 240 个以 1.3GHz 频率运行的核的 Nvidia 280 GTX 显卡上引入新的优化技术扩展了前面的工作，并比较了基于表和基于环的不同实现。分代大小为 128，数据块大小为 32KB 时的解码吞吐量可达 254MB/s。然而，这些参考文献最有趣的贡献是范围广泛的优化技术和在大多统一的硬件平台上比较的不同的方法。它们代表了目前最为完善的加速 RLNC 的 GPU 处理。但是，由于这些实现是基于 Nvidia CUDA 架构的，因此它们只能在 Nvidia 的硬件上运行。

参考文献［23］中给出了一个利用 Nvidia CG 编写并编译为 OpenGL 的实现，其测试配置包括一台使用 Intel Q6600 2.4GHz 四核 CPU 的个人计算机和一个包含 64 个运行在 1.63GHz 频率上的核的 Nvidia 9600GT 显卡。该实现采用的是基于查找表的方法并在显卡的 GPU 上运行。矩阵逆运算通过改进的 Gauss - Jordan 算法实现，域大小是 2^8，分代大小介于 16 ~ 256 之间，编码数据块大小为 1KB。

结果如图 4-11㊀所示。当分代大小为 16 时，解码吞吐量可达 38MB/s，随着分代大小的增加和数据块大小的减少，编码吞吐量显著降低。和前面的工作不同的是，该参考文献中给出的实现可以运行在任何 OpenGL 平台上㊁。这是一个有趣的可能性，因为这可能会引导出在支持 OpenGL 甚至是支持 OpenGL ES 的移动电话上的实现[24]。

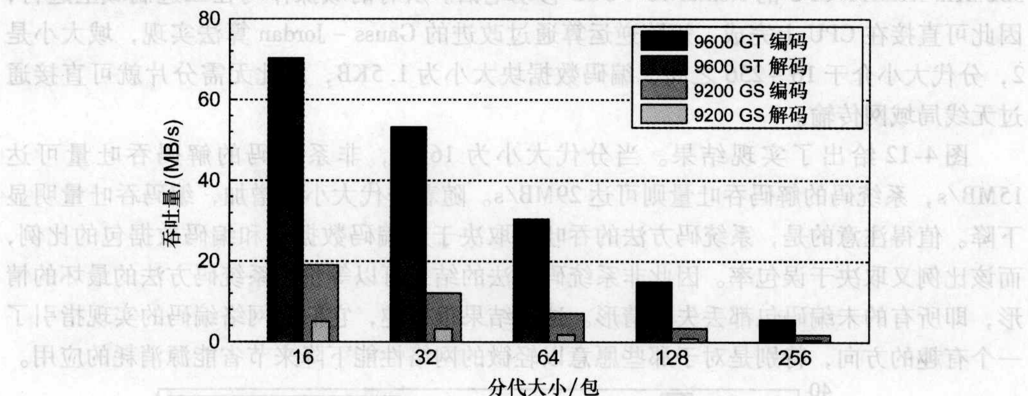


图 4-11　OpenGL 显卡的解码吞吐量。分代大小介于 16 ~ 64 之间，数据块大小为 1KB[23]

4.8　通过设计提升随机线性网络编码

改善编码吞吐量的另一种方法是改变使用的编码设计。本章中我们考虑两种可能的情况：系统码和二进制域上的编码。

使用系统码时，所有的琐碎编码向量㊂都会被使用一次，随后再随机生成编码向量。这导致所有的符号都被不编码地发送一次，之后再生成编码符号并发送。这样做可

㊀ 感谢 Peter Vingelmann 提供此数据。

㊁ 在实际应用中，必要时可以更改/移植代码，以适应不同硬件平台间的不兼容。

㊂ 琐碎编码向量中的标量仅有一项为 1，其他均为 0。

以减少源节点上必须产生的以及汇聚在节点上必须解码的编码符号的数量，从而降低节点上的运算负担。该方法的原理是所有的原始符号都是线性独立的，因此在所有的符号都被发送过一次之前，我们没有必要对它们进行编码。这种方法易于实现并可以减少编码符号的平均数量。

使用二进制域上的编码时，加减法操作可以等价为异或操作，乘法和除法操作虽不需要，但也可以等价为求和（AND）操作。异或操作是所有现有的CPU均可快速完成的基本操作，因而相较于高阶域来说，二进制域上的编解码可以快速实现。随着域大小的减少，任意两个随机生成的编码符号之间线性相关的概率就会增大。然而，正如参考文献［25］表明的，即使在最小的域上编码，线性相关的概率也是令人吃惊的低。因此，如果分代大小足够大，则自由度的损失就无足轻重了。另外，移动电话几乎完全依赖于无线网络，但这类网络的误包率（PEP）一般较高，因此相比于例如10%的高误包率，少量接收到的数据包之间线性相关就不那么重要了。从节能的角度来看，额外传输所增加的成本与编码操作的低复杂度所减少的CPU能源消耗之间存在着权衡关系。

参考文献［25］中给出了一个二进制随机线性网络编码的实现，其平台是使用332MHz ARM11 CPU的NokiaN95－8GB移动电话。所有的域操作均在二进制域上进行，因此可直接在CPU上完成。矩阵逆运算通过改进的Gauss－Jordan算法实现，域大小是2，分代大小介于16～256之间，编码数据块大小为1.5KB，因此无需分片就可直接通过无线局域网传输。

图4-12给出了实现结果。当分代大小为16时，非系统码的解码吞吐量可达15MB/s，系统码的解码吞吐量则可达29MB/s。随着分代大小的增加，编码吞吐量明显下降。值得注意的是，系统码方法的吞吐量取决于未编码数据包和编码数据包的比例，而该比例又取决于误包率。因此非系统码方法的结果可以等价于系统码方法的最坏的情形，即所有的未编码包都丢失的情形。这些结果很有趣，它们为网络编码的实现指引了一个有趣的方向，特别是对于那些愿意以轻微的网络性能下降来节省能源消耗的应用。

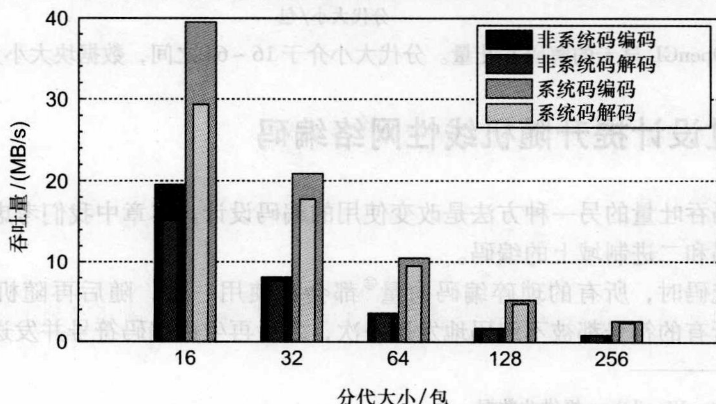


图4-12　在诺基亚N95－8GB上实现的解码吞吐量，分代大小介于16～64之间，数据块大小为1.5KB[25]

4.9 采用了网络编码的移动电话应用

为了说明网络编码在手机上的实用性，我们来考虑参考文献［26］中给出的图片查看器应用。图片查看器应用允许用户将自己手机上的图像广播给一些接收设备，其主要思想是，用户通过短距离技术如WiFi等共享内容，内容可以直接传送到附近的手机而不是上传到诸如MySpace或Facebook等社交网络。该应用允许用户看到自己的所有移动设备上的照片，这比只能看到一台设备上的照片要方便许多。

图片查看器应用允许用户直接监控解码过程，并通过将解码矩阵的内容绘制到接收方的手机显示屏上来显示解码处理过程。该应用提供了一个非常直接的反馈，这对于协议的开发是十分有用的。图4-13a~c中的截图显示的是网络编码的解码过程。

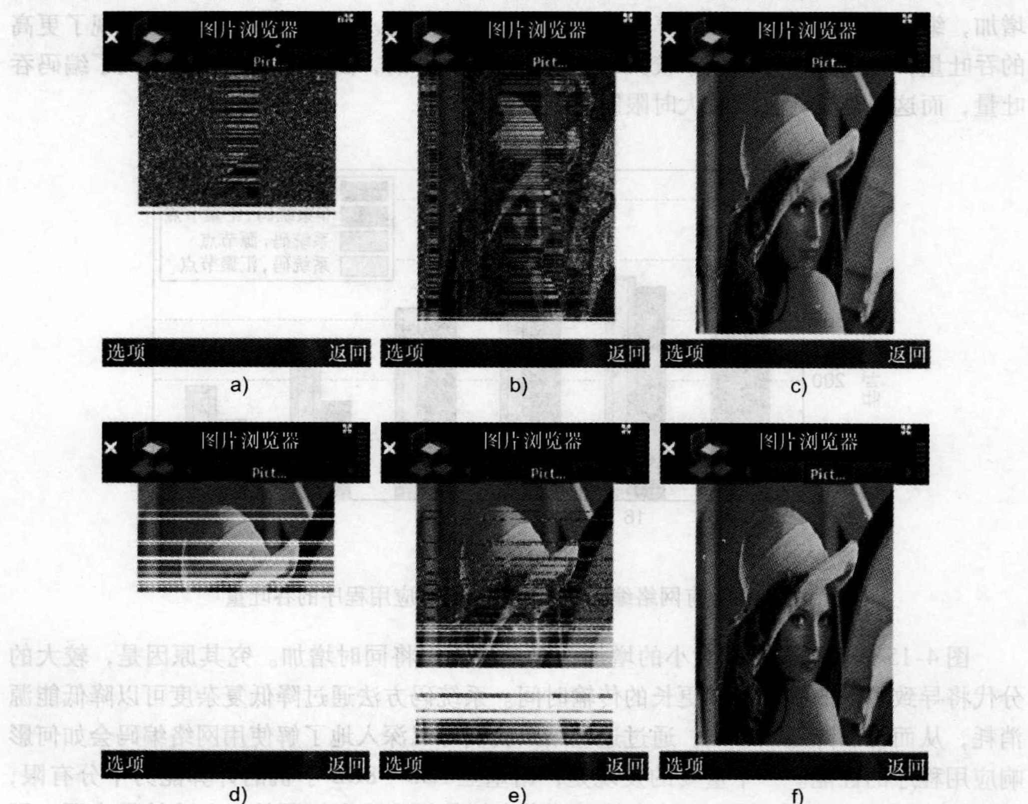


图4-13 纯网络编码

a）部分解码数据 b）随着解码器的秩的增加，图像开始显现 c）最终解码图像。系统网络编码 d）接收到的未编码数据 e）通过编码数据包修复擦除 f）最终解码图像

初始时如图4-13a所示，其解码矩阵的内容看起来像噪声。随着解码器接收到更多的线性组合，解码过程逐步进行，原始图片开始显现，如图4-13b所示。最后在图

4-13c中图片解码完成，传输结束。图4-13d～e中的截图给出了采用系统网络编码的解码过程。图4-13d说明了未编码数据包是如何被插入到解码矩阵中的，该行为将持续直至所有的原始数据包均被发送了一次。图4-13e已进入编码阶段，在这个阶段中，在未编码阶段被擦除的数据包将通过发送的编码数据包被修复。

作者在两台诺基亚N95－8GB移动电话上对该应用进行了测试。其中一部手机作为源节点，需要传输大小为5MB的文件，另一部手机作为汇聚节点。作者对不同的分代大小进行了测试，以说明日益复杂的编码操作是如何影响应用程序的性能的。为了说明网络编码操作对性能的影响，作者以未使用编码时的WLAN的性能作为参考。未编码时的测试结果显示了手机可以播放的最高速度以及广播阶段的功耗。

图4-14表明，随着分代大小的增加，吞吐量将降低。当分代大小较小时，编码的吞吐量与未编码时近似，这表明在这种情况下计算开销不是性能瓶颈。随着分代大小的增加，编码操作需要更强的计算能力并导致吞吐量降低。此外，系统码方法实现了更高的吞吐量，尤其是在分代大小较大时。这是预料之中的，因为系统码方法增大了编码吞吐量，而这正是分代大小较大时限制性能的因素。

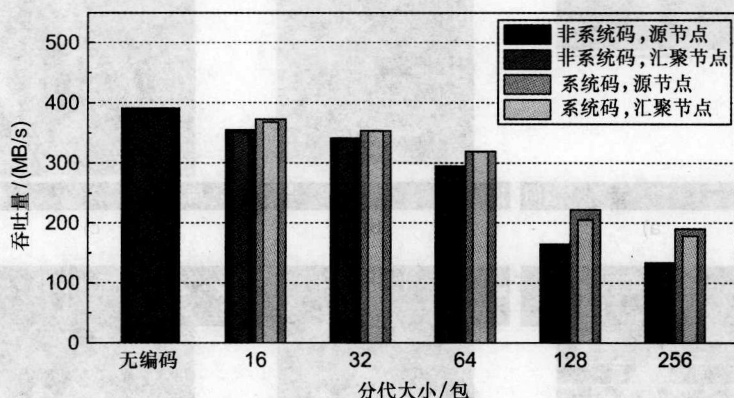


图4-14　有网络编码和无网络编码时应用程序的吞吐量

图4-15表明随着分代大小的增加，能量消耗也将同时增加。究其原因是，较大的分代将导致更低的吞吐量和更长的传输时间。系统码方法通过降低复杂度可以降低能源消耗，从而可以降低该影响。通过这些结果我们可以深入地了解使用网络编码会如何影响应用程序的性能。一个重要的发现是，诺基亚N95－8GB手机的计算能力十分有限，即使是以无线局域网（WLAN）支持的速率广播数据也是有问题的。上述结果表明，即使是低端设备，比如移动电话，其上也可以部署网络编码。希望未来的实际实验可以揭示哪些应用程序和网络拓扑结构可以通过网络编码来提高性能。

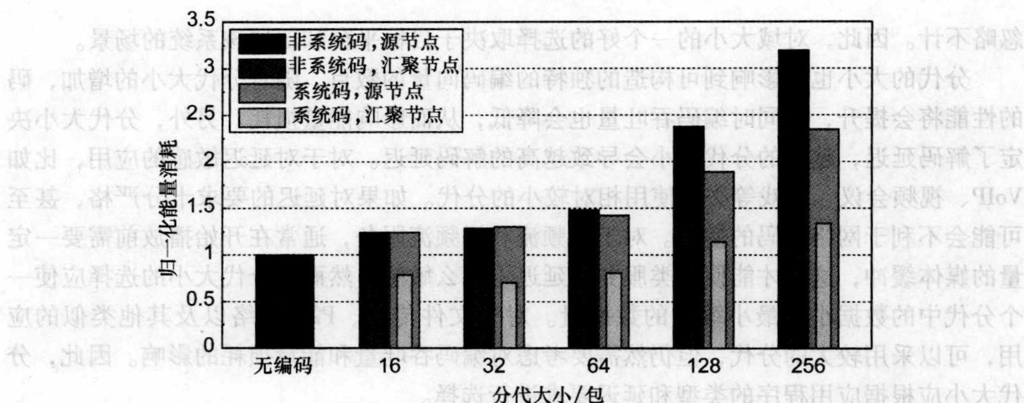


图 4-15 有网络编码和无网络编码时应用程序的归一化能量消耗

4.10 隐患和参数

网络编码系统中的节点必须商定至少 3 个参数，分别是域大小、分代大小和数据块大小。图 4-16 中显示了对域大小和分代大小的选择带来的影响。这些参数可以预先设定或作为边信息被包含在每个编码数据片中。不存在对这些参数的一般性的最优选择，因为它们依赖于若干因素，比如目标平台、网络技术和应用程序。然而，在实现和测试时切合实际的选择是值得考虑的，例如避免对不会在现实应用中使用到的设置进行优化。因此下面我们将给出这些选择以及不同的应用程序可能面对的要求。

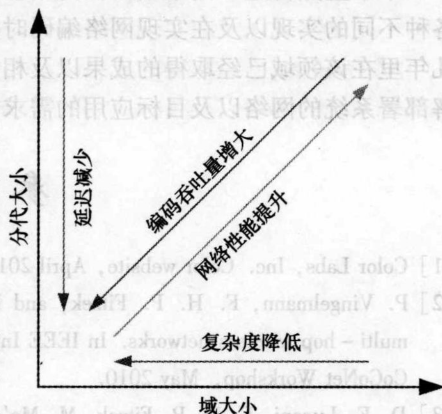


图 4-16 不同的域大小和分代大小之间的折衷关系

域的大小是一个重要的参数，因为它直接影响到可构造的独特的编码向量的数目。特别是，随机线性网络编码所使用的域的大小决定了接收到的组合线性相关的概率，这反过来又决定了码的性能。目前最常用的域大小是 2^8，该取值可以在码的性能和编码吞吐量之间实现一个合理的权衡。此外，当基于查找表方法实现时，由于目前所有的平台实现的数据类型都是这一规模，因此这一域的大小是实际的。当域的大小为 2 时，可以提供更高的编码吞吐量，但也将略微恶化接收数据包之间线性独立的概率。域大小的选择应根据目标平台和场景确定。对于误包率达到 10% 的无线场景，在一个较大的域上执行操作所需的能量可能会多于发送一些额外数据包的能量，因此一些额外的线性相关的数据包是可以接收的。但如果我们考虑的是 PEP 接近于 0% 的有线情况，且所有的节点都具备一个强大的 CPU，则选择一个较大的域的成本⊖可以

⊖ 在能量和计算资源方面。

忽略不计。因此，对域大小的一个好的选择取决于目标平台和部署该系统的场景。

分代的大小也会影响到可构造的独特的编码向量的数目。随着分代大小的增加，码的性能将会提升，但同时编码吞吐量也会降低，从而影响能量损耗。另外，分代大小决定了解码延迟，越高的分代大小会导致越高的解码延迟。对于对延迟敏感的应用，比如VoIP、视频会议、游戏等必须使用相对较小的分代。如果对延迟的要求十分严格，甚至可能会不利于网络编码的使用。对于音频流和视频流服务，通常在开始播放前需要一定量的媒体缓冲，这样才能使这类服务对延迟不那么敏感。然而，分代大小的选择应使一个分代中的数据小于最小缓冲的数据量。对于文件传输、P2P 网络以及其他类似的应用，可以采用较大的分代，但仍然需要考虑对编码吞吐量和能量损耗的影响。因此，分代大小应根据应用程序的类型和延迟要求进行选择。

数据块的大小或数据包的大小同样也会显著地影响编码吞吐量，尤其是对于并行实现的情况，采用较大的数据块可能是实现最佳吞吐量所必需的。请注意，为了保证块的可用性，编码块中的所有数据包必须按顺序被接收。还要注意，当数据块较大时，它将会被分割到网络的若干个帧中。这种情况下，若丢失一个帧，剩余的帧将是无用的。因此，我们应基于发送编码包的网络技术以及实施细节来定义块的大小。

本章重点讲述了网络编码在商用移动平台上的实现。我们希望本章有助于读者了解各种不同的实现以及在实现网络编码时必须作出的设计选择。此外，本章还概述了过去几年里在该领域已经取得的成果以及相关参考文献。最后我们要强调，最重要的是要理解部署系统的网络以及目标应用的需求。

参考文献

[1] Color Labs, Inc. Color website, April 2011. http://color. com.

[2] P. Vingelmann, F. H. P. Fitzek, and D. E. Lucani. Application - level data dissemination in multi - hop wireless networks. In IEEE International Conference on Communications (ICC 2010) - CoCoNet Workshop, May 2010.

[3] D. E. Lucani, F. H. P. Fitzek, M. Me'dard, and M. Stojanovic. Network coding for data dissemination: It is not what you know, but what your neighbors know. In RAWNET/WNC3 2009, June 2009.

[4] R. Jacobsen, K. Jakobsen, P. Ingtoft, T. Madsen, and F. H. P. Fitzek. Practical Eval - uation of Partial Network Coding in Wireless Sensor Networks. In 4th International Mobile Multimedia Communications Conference (MobiMedia 2008), Oulu, Finland, July 2008. ICTS/ACM.

[5] D. Traskov, J. Lenz, N. Ratnakar, and M. Me'dard. Asynchronousnetwork coded multicast. In IEEE International Conference on Communications (ICC 2010), May 2010.

[6] P. Sadeghi, R. Shams, and D. Traskov. An optimal adaptive networkcoding scheme for minimizing decoding delay in broadcast erasurechannels. In EURASIP Journal of Wireless Communications and Networking, Special Issue on Network Coding for Wireless Communications, pages 1 - 14, 2010.

[7] Donald E. Knuth. The art of computer programming, volume 2 (3rd ed.): Seminumerical algo-

rithms. Addison - Wesley Longman Publishing Co. , Inc. , Boston, MA, USA, 1997.

[8] Takuji Nishimura and Makoto Matsumoto. Mersenne twister home page. http://www. math. sci. hiroshima - u. ac. jp/~m - mat/MT/emt. html.

[9] Cheng Huang and Lihao Xu. Fast software implementations of finite field operations. Technical report, Department of Computer Science & Engineering, Washington University, St. Louis, MO 63130, December 2003. Available online: http://nisl. wayne. edu/Papers/Tech/GF. pdf.

[10] Neal R. Wagner. The Laws of Cryptography with Java Code. 2003. Unpublished, available online: http://www. cs. utsa. edu/~wagner/lawsbookcolor/laws. pdf.

[11] Lieven Vandenberghe. Applied numerical computing. http://www. ee. ucla. edu/~vandenbe/103/reader. pdf.

[12] Morten V. Pedersen, Frank H. P. Fitzek, and Torben Larsen. Implementation and performance evaluation of network coding for cooperative mobile devices. In IEEE Cognitive and Cooperative Wireless Networks Workshop. IEEE, May 2008.

[13] S. Katti, H. Rahul, W. Hu, D. Katabi, M. M'edard, and J. Crowcroft. XORs in the air: practical wireless network coding. In Proceedings of the 2006 conference on applications, technologies, architectures, and protocols for computer communications (SIGCOMM'06), pages 243 - 254. ACM Press, September 11 - 15, 2006.

[14] MeaWang and Baochun Li. How practical is network coding? Quality of Service, 2006. IWQoS 2006. 14th IEEE International Workshop on, pages 274 - 278, June 2006.

[15] Mea Wang and Baochun Li. Lava: A reality check of network coding in peer - to - peer live streaming. In INFOCOM, pages 1082 - 1090, May 2007.

[16] Szymon Chachulski, Michael Jennings, Sachin Katti, and Dina Katabi. Trading structure for randomness in wireless opportunistic routing. In SIGCOMM '07: Proceedings of the 2007 conference on applications, technologies, architectures, and protocols for computer communications, pages 169 - 180. ACM, August 2007.

[17] Janus Heide, Morten V. Pedersen, Frank H. P. Fitzek, and TorbenLarsen. Cautious view on network coding - from theory to practice. Journal of Communications and Networks (JCN), 10 (4): 403 - 411, December 2008.

[18] Hassan Shojania and Baochun Li. Random network coding on the iphone: Fact or fiction? In NOSSDAV '09: Proceedings of the 18th international workshop on network and operating systems supportfor digital audio and video, pages 37 - 42. ACM, June 2009.

[19] Hassan Shojania and Baochun Li. Parallelized progressive network coding with hardware acceleration. In Fifteenth IEEE International Workshop on Quality of Service, pages 47 - 55, June 2007.

[20] Xiaowen Chu, Kaiyong Zhao, and Mea Wang. Massively parallel network coding on GPUs. In Performance, Computing and Communications Conference, 2008. IPCCC 2008. IEEE International, pages144 - 151, December 2008.

[21] Hassan Shojania, Baochun Li, and Xin Wang. Nuclei: GPU - accelerated many - core network coding. In The 28th Conference on Computer Communications (INFOCOM 2009), April 2009.

[22] Hassan Shojania and Baochun Li. Pushing the envelope: Extreme network coding on the GPU. In

ICDCS, pages 490 – 499, June 2009.

[23] Peter Vingelmann, Peter Zanaty, Frank H. P. Fitzek, and Hassan Charaf. Implementation of random linear network coding on openglenabled graphics cards. In European Wireless 2009, May 2009.

[24] The Kronos Group. Kronos website, May 2011.

[25] Janus Heide, Morten V. Pedersen, Frank H. P. Fitzek, and Torben Larsen. Network coding for mobile devices – systematic binaryrandom rateless codes. In The IEEE International Conference on Communications (ICC), Dresden, Germany, 14 – 18 June 2009.

[26] Morten V. Pedersen, Janus Heide, Frank H. P. Fitzek, and Torben Larsen. Pictureviewer – a mobile application using network coding. In The 15th European Wireless Conference (EW), Aalborg, Denmark, 17 – 20 May 2009.

第5章 网络编码和用户协同在LTE网络流和下载服务上的应用

Qi Zhang 丹麦，奥尔胡斯，奥尔胡斯大学

Janus Heide, Morten V. Pedersen, Frank H. P. Fitzek 丹麦，奥尔堡，奥尔堡大学，电子系统系

Jorma Lilleberg, Kari Rikkinen 芬兰，奥卢，瑞萨通信技术公司

摘要

本章将讲述网络编码和用户协同在现有LTE网络中的应用。LTE网络利用Raptor编码来实现内容分发，比如下载和流服务，并通过空中传送给移动设备。然而，上述方法存在一些缺点。与其他前向纠错编码（FEC）技术一样，Raptor编码引入的冗余会增加系统总带宽。由于它需要不断地修复信道特性较差的移动设备的消息，因此会增大信道特性较好的移动设备的能量消耗。此外，Raptor码会大幅增加每个用户的感知延迟，这可能会产生问题，特别是对于流媒体服务。需要为了克服以上描述的带宽和能耗问题，我们提出了使用网络编码的用户协同来支持未来手机通信系统中的流服务和下载服务。仿真结果表明，只要有至少两个移动设备进行协同传输，本地重传就可以节省蜂窝链路上高达80%的冗余信息。仿真结果还表明，当协同集群中有4台设备时，网络编码可以节省短距离链路上的一半以上的流量。

关键字

用户协同、LTE、Raptor码、短距离通信。

5.1 引言

随着诸如视频流、软件分发、本地新闻、天气预报等未来移动设备业务对多元化带宽需求的增加，蜂窝网络仅使用传统的点到点（PTP）传输方法已无法满足所有的需求。幸运的是，流行的业务常常拥有很多用户。一种更高效的内容传送方法可以解决这个问题，我们称之为点对多点传输（PMP）。这一趋势已被第三代合作伙伴计划（3GPP）确认[1]，并且从3GPP的版本6之后多媒体广播/组播服务（MBMS）已经被标准化[2]。3GPP的版本9在长期演进（LTE）通信系统中引入了演进的多媒体广播组播服务（eMBMS）。相比于MBMS，eMBMS使用OFDM组播/广播单频网络提高性能，提升了信干比（SINR），并增加了吞吐量。

同MBMS一样，eMBMS支持两种类型的多媒体内容分发业务，即流服务和下载服务。流服务用来分发连续的视频和音频数据流到移动设备，而下载服务则用于分发二进制数据、静止图像、文本或软件版本。因两种业务具有不同的特性，eMBMS采用不同

的方法处理流和下载服务。需要特别指出的是，由于流服务有严格的延迟限制，移动设备没有时间发送 ACK 或 NACK 等用于纠删的反馈信息给基站，因此该传输可看做是没有交互重传的单向通信，其纠删只能依靠端到端的纠删码，比如 Raptor 码[5]。考虑到没有码字能完全保证及时无误地进行数据分发，因此我们通常假设端用户可以忍受偶尔的译码错误。相反地，下载业务则没有特别的延迟要求，但它要求所有移动设备接收到的数据都是零误码，因此必须重传丢失或无法解码的数据包，以保证可靠传输。

为了满足上述两种业务的需求，eMBMS 进程包括三个阶段[3]：

- 初始化阶段：使用 TCP/IP 上的双向 PTP 通信或单向 PMP 传输来宣布并建立 eMBMS 业务。
- 分发阶段：由使用 IP 上的 UDP 的协议支撑的单向 PMP 传输来传输流或下载服务，例如 RTP（实时传输协议）和 FLUTE（使用单向传输的文件传送）。
- 后分发阶段（可选）：使用 TCP/IP 上的双向 PTP 通信来报告所接收内容的质量，并确保每个用户都接收到了整个文件。

图 5-1 中给出了 eMBMS 的架构及其关键组成部分。该架构不仅支持传统的单小区广播，还引入了新的特性——MBMS 单频网络（MBSFN）。MBMSFN 的基本思想是：多个 eNB（演进节点）通过同一个 MBMS 信道以时间同步的方式将多媒体内容广播/组播给它们的 UE（用户设备）。MBSFN 承诺使用效率更高的方式将多媒体内容分发给 UE，其频谱效率更高。eMBMS 网关（eMBMS GW）负责将流量分配给 MBSFN 区域中的不同的 eNB。eMBMS 网关包括用户平面和控制平面。eMBMS 网关通过用户平面参考点 M_1 与 eNB 通信进行 eMBMS 数据分发。同时，eMBMS 网关通过控制平面参考点 M_3 与 MCE（组播协调实体）进行通信。对于给定的业务，MCE 用于确保将同一资源块分配给给定 MBSFN 区域中的所有 eNB。M_2 是 E－UTRAN（演进的 UMTS 陆地无线接入网络）中的内部控制平面接口。图 5-1 中的 MCE 是一个单独的实体，但它也可以是 eNB 的一部分。

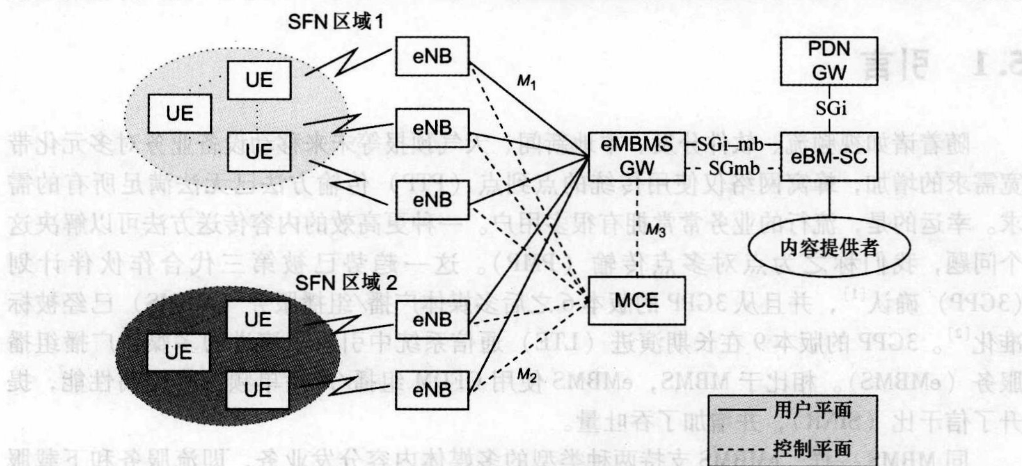


图 5-1 eMBMS 网络架构

SGi - mb 表示用于 eMBMS 数据传输的 eBM - SC（演进的广播/多播服务中心）和 eMBMS 网关之间的参考点。SGmb 是 eBM - SC 和 eMBMS 网关之间在控制平面上的参考点。eBM - SC 通过 SGi 接口与分组数据网络网关（PDN GW）连接。

eMBMS 中最重要的挑战之一就是纠错。不论是流业务还是下载业务，eMBMS 都是用来将数据分发给多个 UE。由于无线多播网络中数据包丢失的复杂性，不同的 UE 上发生的数据包丢失可能相关也可能独立，详细内容我们将在 5.3 节中讨论。此外，对于流服务，在 eNB 上使用传统的重传方案是不可行的。虽然在下载服务中 eNB 可以通过后交付阶段来修复丢失的数据包，但当 UE 的数量较大时，后交付阶段也会消耗大量的无线资源。因此，eMBMS 必须采用先进的前向纠错技术。在 eMBMS 中，FEC 被同时应用于物理层和应用层，在物理层上采用 3G 空中接口信道编码方案纠正比特错误，在应用层上采用 Raptor 码恢复丢失的数据包。

Raptor 编码[5]是一种新型的纠删编码技术。Raptor 码通过引入预编码处理对 LT 码[4]进行了扩展，目的是提高解码概率。Raptor 码是目前最有效的纠删码之一。然而，Raptor 码方案与延迟、带宽占用和纠删性能之间的权衡有关。

纠删码把 k 个源符号编码为 N 个编码符号（$N>k$），因此我们可以通过编码符号的子集来重构源符号。为了在 eMBMS 中实现 Raptor 码，我们需要考虑 k 和 $N-k$ 两个参数。参数 k 决定了源数据块的大小。源数据块的大小会对延迟造成影响，在流服务中它应取较小的值。另一方面，当源数据块较大时，Raptor 码将获得更好的性能。其次，额外的修复符号的数量决定了实现 FEC 需要的额外带宽。修复信息越多，设备恢复丢失数据包的可能性就越大。但消极的一面是，发送大量的修复符号需要消耗大量的蜂窝带宽。此外，已经正确接收到信息的设备需要保持空闲状态直至下一个传输阶段，由于这些设备在空闲状态也需要监测信道，因此会造成额外的能量消耗。

本章将讨论 LTE 网络中提高流服务和下载服务的方法。我们的目标是通过引入用户协同和网络编码来最小化服务延迟并降低带宽占用。为了符合标准的 3GPP 规范，我们在蜂窝链路上采用 Raptor 编码。除蜂窝链路以外，用户协同还可以利用在距离相近的移动设备之间使用诸如蓝牙和 WiFi 等短距离通信技术建立的本地链路。这种情况下，移动设备不是单独行动而是形成一个协同小组，我们称之为无线网格。相比于独立的设备，协同集群在频谱效率、能源消耗和可实现数据速率等方面具有许多优势（参见参考文献［6，7，9-11］）。本章中用户协同被用于协同集群内的本地重传，以减少蜂窝链路的负担。集群内的数据包交换通过网络编码技术实现，以降低整个集群中的必要传输。

本章安排如下：5.2 节描述了 Raptor 码的基本理论并给出了 eMBMS 中与 Raptor 码实现相关的权衡。5.3 节阐述了无线组播场景下的包删除图样。5.4 节提出并描述了用于恢复包删除的用户协同。5.5 节介绍了如何将网络编码应用到用户协同本地重传中，以提高本地重传效率。5.6 节给出了仿真结果。5.7 节对本章进行了总结。

5.2 eMBMS 中的 Raptor 码

Raptor 码在应用层提供数据包级保护，这对物理层的比特级 FEC 是一个补充。在演进的广播多播服务中心（eBMSC）上和所有独立的移动设备上均已实现了 Raptor 码。其基本编码过程如下：把一个完整的数据文件或一段数据流插入到一个大数据块中，我们称之为源块。每个源块包含有 k 个符号，每个符号又包含有 T 个字节，也就是每个源块包含有 kT 个字节。图 5-2 中给出了源块的结构。



图 5-2　源块的结构

一个完整的源块构建完成后，Raptor 编码器将生成 $N-k$ 个大小为 T 字节的修复符号。修复符号的数目，即 $N-k$ 取决于源块是用于 eMBMS 中的流服务还是下载服务、参与网络条件、期望服务质量、有效的额外带宽以及允许的传输时间[3]。eMBMS 采用系统码，即先发送 k 个源符号，接着再发送 $N-k$ 个修复符号给所有的接收者。

系统码的好处是，一些接收到的源符号无需解码就可以直接被接收机使用。即使是在无法解码源块的情形下，前 k 个符号中的原始数据仍可以被传递给高层，并在高层通过错误隐藏算法来处理错误。在接收端，接收机只要接收到足够的数据包（源包或修复包），就可以解码出源块。如果接收机由于多个数据包丢失而无法接收到足够多的数据包来解码整个源块，那么它就只把被解码的部分源块传递给高层。

图 5-3 给出了 eMBMS 中的流媒体和下载服务的总体框架示意图。流服务和下载服务采用了相似的架构。其主要区别是：流媒体应用采用 RTP（实时传输协议）传输数据，

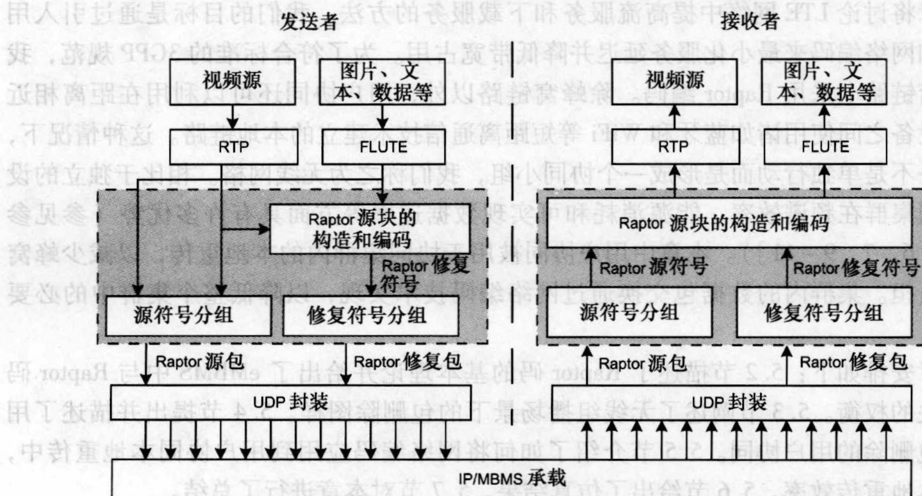


图 5-3　eMBMS 中的流服务和下载服务的框架

而非实时应用则采用FLUTE（单方向文件分发协议）传输数据。参考文献［3］中给出了有关RTP和FLUTE的详细描述。

为了选择合适的源块大小k和修复符号数目$N-k$，需要综合考虑源块大小、延迟、期望服务质量、修复符号和带宽扩展之间的权衡关系。

我们以流服务作为实例。流服务中常使用平均失败间隔时间（MTBF）来衡量期望服务质量。假设每个源块包含t_s时间的视频内容，解码失败的概率为P_{df}，则解码失败间隔时间的期望值为t_s/P_{df}。

要满足MTBF需求τ，下式必须成立：

$$\frac{t_s}{P_{df}} \geqslant \tau \tag{5-1}$$

Raptor码解码失败的概率为[12]

$$P_{df} = \begin{cases} 1 & m < k, \\ 0.85 \times 0.567^{m-k} & m \geqslant k \end{cases} \tag{5-2}$$

其中，P_{df}表示当对k个源符号编码时，接收到m个符号后解码失败的概率[12]。

对于给定的预期包错误率p_e，假设eBMSC对于每个源块都发送N个符号，且每个符号都被封装为一个数据包，则成功接收的符号数量的期望值可以表示为

$$m = N\ (1-p_e) \tag{5-3}$$

根据式（5-1）～式（5-3）可知，为了满足可靠性要求，每块包含的符号数目须至少为

$$\begin{aligned} N &\geqslant \frac{1}{1-p_e}\left(\log_{0.567}\frac{t_s}{0.85\tau} + k\right) \\ &\geqslant \frac{1}{1-p_e}\left(k - 1.7624\ln\frac{t_s}{\tau} - 0.2864\right) \end{aligned} \tag{5-4}$$

因此，每块包含的修复包的最小数目R为

$$\begin{aligned} R &= N - k \\ &= \frac{1}{1-p_e}\left(p_e \cdot k - 1.7624\ln\frac{t_s}{\tau} - 0.2864\right) \end{aligned} \tag{5-5}$$

如果视频流的数据速率为$r$bit/s，则感知延迟D可以表示为

$$D = t_s = \frac{8kT}{r} \tag{5-6}$$

根据式（5-6）不难看出，较大的源块将导致较长的感知延迟。例如，假设目标MTBF为1h，视频流媒体播放数据速率为512kbit/s，符号大小T为256B，平均误包率为10%。对于上述配置，表5-1中列出了不同源块大小下的开销和延迟，这里的开销定义为冗余包数目和信息包数目的比值，即R/k。从表5-1中可以看出，延迟随源块大小的增加而线性增加，相反地，开销随源块大小的增加而减少。因此，要满足延迟限制最好采用较小的源块。而另一方面，要实现同样的性能，使用较小的源块时所需的修复符号数量大于使用较大的源块时所需的修复符号数量。

此外，表5-1中列出的开销是在假设已知丢包率时的理想理论值。实际中无线组播网络的包丢失图样受各种因素的影响，比如信道条件和接收机的硬件质量，是难以预料的（相关的详细讨论见5.3节）。一个数据块内的平均误包率都可能会变化，不同数据块的平均误包率可能完全不同，尤其是当数据块较小时。下节我们将讨论包丢失图样对Raptor码性能的影响。

以上是对于流服务的举例。对于下载服务来说，由于它要求无差错接收，因此其开销更加显著。eNB需要在后分发阶段重传丢失的符号，这将花费额外的时间和带宽。

表5-1 不同源块长度下的开销和延迟

源块长度/B	1024	2048	8192
开销	0.1238	0.1168	0.1122
延迟/s	4.0960	8.1920	32.7680

5.3 包丢失图样

无线多播网络中的不同的移动设备具有不同的数据包丢失图样。包丢失可以由多种原因引发，例如网络拥塞、深衰落、严重的路径损耗、干扰和硬件性能。许多理论研究工作往往都低估了硬件性能的影响。

组播无线网络中的数据包丢失可以相关和（或）不相关。相干时间[㊀]内的包丢失通常是相关的。然而，移动设备上的数据包丢失大多是独立的，每个移动设备的包丢失行为都是异构的，其主要原因是由于节点位置的不同以及制造工艺的不同。在eMBMS中，不同的移动设备位于同一小区的不同位置甚至是不同的小区，这将导致不同的接收机经历不同的路径损耗、衰落、接收干扰等。因此，不同移动设备之间的信道条件大相径庭。此外，参考文献［13-15］中给出的许多实际测量结果揭示了一个有趣的事实：即使是具有相同软硬件的移动设备，其性能也存在很大的区别。这是由于显著的制造差异造成的。参考文献［14］中使用诺基亚N95手机在网络层的测量表明了数据包错误概率的显著变化，其中的一些设备在所有的测试中都表现出了较高的数据包错误概率。这些调查结果在参考文献［13］中得到了验证，在相同的测量条件下使用RT2500芯片组的不同卡之间的帧错误率差异显著。

简单地说，eMBMS中的所有移动设备的数据包丢失图样是相关丢失、独立丢失、随机丢失和突发丢失的组合。

在5.2节中提到过数据块越大，Raptor码的性能越好。其原因之一是，根据大数定律，数据块越大，每块丢失的数据包的数量接近于平均值的概率越高。这意味着，Raptor码产生的修复符号的数目越接近于预测值，则满足预期QoS的概率就越高。然而，当数据块较小时，每块丢失的数据包的数目会在较大的范围内变化，这意味着Raptor码产生的修复数据包的数目可能不足以达到期望的QoS级别。

㊀ 相干时间是指信道幅度从它的前一个值变为一个不相关值所需的最小时间。

图 5-4 所示的是不同的源块长度下，实际数据包错误率的累积分布函数。假设数据包的平均错误率是 10%，我们分别仿真了源块长度（SBL）为 1024B、2048B 和 8192B 时的 1000 个块的传输。研究结果表明，三种情况下 99% 的块的数据包错误率分别小于 12.1%、11.4% 和 10.6%。因此，为了要使 99% 的数据块都可以解码，当 SBL 为 1024B 时，Raptor 码的编码器需要针对 12.1% 的包错误率生成修复符号；当 SBL 为 8192B 时，需要针对 10.6% 的包错误率生成足够多的修复符号。因此，修复符号引入的实际开销要大于其理论值。

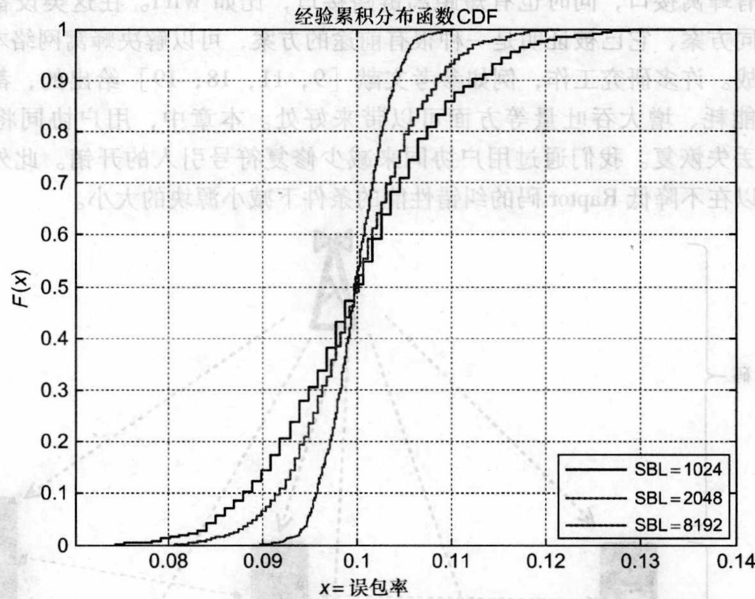


图 5-4　不同源块长度下每块的实际数据包错误率的累积分布函数

当组播组中有两个或两个以上的移动设备时，我们须将其中情况最差的移动设备的误包率作为系统的误包率。例如，如果有 4 个移动设备分别在 SBL 等于 1024B、2048B 和 8192B 的情况下接收组播内容，不同 SBL 下的 4 个节点中实际最高的误包率分别为 12.7%、11.8% 和 10.95%。表 5-2 中对实际得到的结果和理论值进行了比较。表中的值表明，数据块较小时的开销大于数据块较大时的开销。该例中，在单节点情形下，当 SBL 为 1024B 时的开销是 24.7%，这一值要大于当 SBL 为 8192B 时的开销。此外，该表还表明在 4 节点情形下的开销要大于单节点情形下的开销。该表指出，当移动设备的数量增加时，开销也将增大。值得一提的是，1024B 是 3GPP[16] 中定义的最小的源块大小，这主要是因为源块小于 1024B 时的开销很大，以至于会影响效率。

表 5-2　不同源块长度下的理论开销和实际开销的比较

源块长度/B	1024	2048	8192
开销（理论）	0.1238	0.1168	0.1122
开销（实际 - 1 节点）	0.1493	0.1344	0.1197
开销（实际 - 4 节点）	0.1585	0.1396	0.1241

5.4　基于用户协同的丢失恢复

用户协同的概念是在参考文献［6，17］中提出的。用户协同的基本思想是：移动设备与其附近的邻居设备通过短距离链路交换数据包，以实现一个个体的或共同的目标。图5-5给出了用户协同的总体网络架构。由于大多现有的移动设备都是多模设备，即它们不仅有蜂窝接口，同时也有短距离链路接口，比如WiFi。在这类设备中有机会实现用户协同方案，它已被证明是一种很有前途的方案，可以解决蜂窝网络本身无法处理的许多挑战。许多研究工作，例如参考文献［9，11，18，19］给出的，都表明用户协同在节约能耗、增大吞吐量等方面可以带来好处。本章中，用户协同将被应用于eMBMS中的丢失恢复。我们通过用户协同来减少修复符号引入的开销。此外，用户协同的引入可以在不降低Raptor码的纠错性能的条件下减小源块的大小。

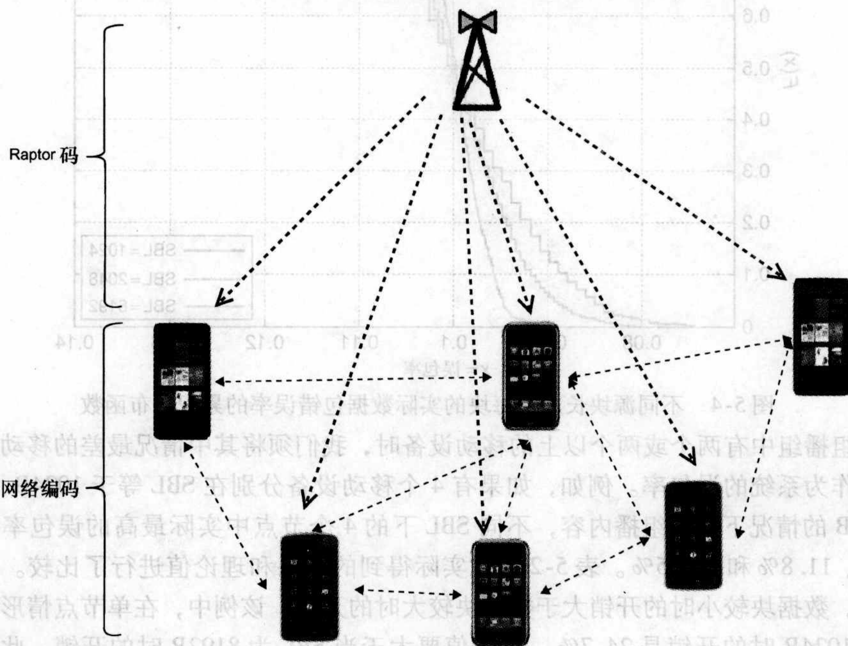


图5-5　用户协同的网络架构

正如我们在5.3节中提到的，包丢失图样是组播无线网络中许多因素的组合造成的，但移动设备之间的不相关的包丢失在所有丢失的数据包中占主导地位。这意味着与某移动设备相邻的设备往往包含有该移动设备丢失的数据包。换句话说，移动设备之间可以通过用户协同来互相帮助，以恢复丢失的数据包。

在采用Raptor码的eMBMS中，该协同丢失恢复方案的工作原理如下。当发送端发出某块中的一批编码数据包后，它需要花几秒钟时间来完成该块中所有数据包的发送。接收端将同样需要几秒钟的时间接收足够多的数据包来构建解码块。Raptor码的编解码

过程在 3GPP TS26.348[8]中有详细介绍。在提出的协同丢失恢复方案中，移动设备没有必要等到块的结尾才开始丢失恢复。移动设备可以在本地逐步进行丢失恢复。例如，本地恢复可以每 64 个数据包进行一次，这不仅减少了 eNB 需要发送的修复符号的数量，而且还允许快速获得大量的修复符号，进而减少了终端节点的解码时间。图 5-6 所示的是逐步恢复的过程。

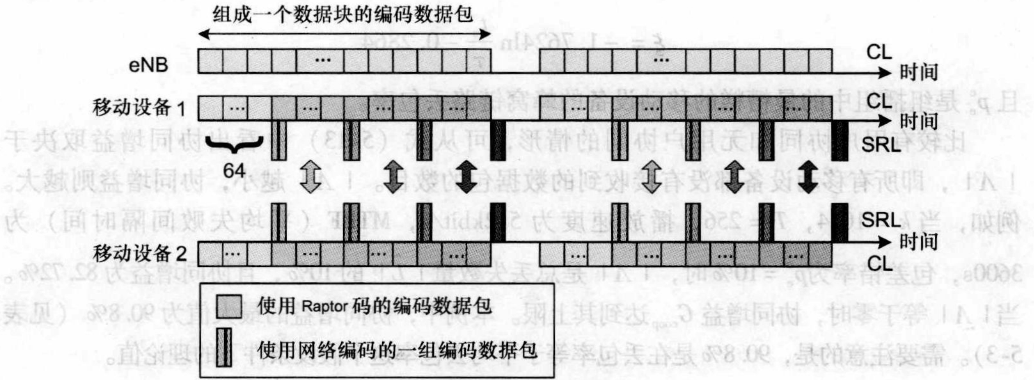


图 5-6　用户协同逐步恢复示意图

在利用用户协同的本地重传中，集群中的所有丢失数据包的集合可以表示为

$$\hat{L} = L_1 \cup L_2 \cdots \cup L_n \tag{5-7}$$

其中，L_1，L_2，…，L_n 分别是移动设备 1，2，…，n 所丢失的数据包的集合。

集群中的公共丢失数据包的集合可以表示为

$$\Lambda = L_1 \cap L_2 \cdots \cap L_n \tag{5-8}$$

当 $\Lambda = \phi$ 时，集群中所有的丢失数据包都可以恢复。可通过本地重传恢复的数据包的总数 N_{rv} 表示为

$$N_{rv} = |\hat{L}| - |\Lambda| \tag{5-9}$$

其中，$|\hat{L}|$ 是集合 $\hat{L}$ 的基数，也就是移动设备的总丢包数；$|\Lambda|$ 是该集群的丢包数。

利用本地重传，eBMSC 可为每个源块发送 N_{coop} 个符号，这些符号包括了源符号和修复符号。则成功接收的符号数目 m_{coop} 可以表示为

$$m_{coop} = N_{coop} - |\Lambda| \tag{5-10}$$

根据式（5-1）、式（5-2）和式（5-10）可知，由 eBMSC 发送的符号的最小数量可由下式得到：

$$N_{coop} \geqslant k + |\Lambda| - 1.7624\ln\frac{t_s}{\tau} - 0.2864 \tag{5-11}$$

利用用户协同，每个源块的修复符号的最小数目 R_{coop} 则为

$$\begin{aligned} R_{coop} &= N_{coop} - k \\ &= |\Lambda| - 1.7624\ln\frac{t_s}{\tau} - 0.2864 \end{aligned} \tag{5-12}$$

在蜂窝链路中，基于用户协同的本地重传可以将每个数据块所需的修复符号的最小数目从 R 减小到 R_{coop}。因此，用户协同增益 G_{coop} 可以表示为

$$G_{coop}=\frac{R-R_{coop}}{R}=1-\frac{(1-p_e^c)(|\Lambda|+\xi)}{p_e^c\cdot k+\xi} \tag{5-13}$$

其中

$$\xi=-1.7624\ln\frac{t_s}{\tau}-0.2864$$

且 p_e^c 是组播组中的最糟糕的移动设备的蜂窝链路丢包率。

比较有用户协同和无用户协同的情形，可从式（5-13）中看出协同增益取决于 $|\Lambda|$，即所有移动设备都没有接收到的数据包的数目。$|\Lambda|$ 越小，协同增益则越大。例如，当 $k=1024$，$T=256$，播放速度为512kbit/s，MTBF（平均失败间隔时间）为3600s，包差错率为$p_e^c=10\%$时，$|\Lambda|$ 是总丢失数量 $|\hat{L}|$ 的10%，且协同增益为82.72%。当 $|\Lambda|$ 等于零时，协同增益 G_{coop} 达到其上限。本例中，协同增益的最大值为90.8%（见表5-3）。需要注意的是，90.8%是在丢包率等于平均丢包率这个假设条件下的理论值。

表5-3　协同增益

$\lvert\Lambda\rvert$	0	$0.1\hat{L}$	$0.2\hat{L}$	$0.3\hat{L}$	$0.4\hat{L}$	$0.5\hat{L}$
G_{coop}	0.9080	0.8272	0.7464	0.6656	0.5848	0.5040

此外，使用基于用户协同的本地重传时，下载服务在后交付阶段可以节省大量的重传流量，这是因为集群中的大部分丢包可以通过短距离链路恢复。另外，由于短距离链路的数据传输速率通常高于蜂窝链路，因此移动设备只需花费较少的时间就可以恢复所有的丢包。

通过使用短距离链路实现的协同增益是非常可观的。但是，使用短距离链路也会带来一些相关的成本。这些成本可以分成两个部分：移动设备的通信成本和网络资源成本。

移动设备的通信成本主要是指花费在短距离通信上的能耗。短距离链路的每比特能量比值（EpBR）要比蜂窝链路小很多。因此，与在蜂窝链路上节省的能量相比，这些额外的能量成本很小。换而言之，即使用户协同会在短距离链路上带来额外的能源成本，移动设备在整体能耗上仍然能够获得显著的节能增益。关于在无线网络中利用用户协同以减少能源消耗的详细讨论参见参考文献［9，11，18，19］。

为了评估网络资源成本，我们需要计算通过短距离链路交换的数据包的数量 N_{sr}，该数量可以表示为

$$N_{sr}=\frac{N_{rv}}{1-p_e^{sr}}=\frac{|\hat{L}|-|\Lambda|}{1-p_e^{sr}} \tag{5-14}$$

其中，p_e^{sr} 是短距离链路的误包率，请注意，在实际中 p_e^{sr} 小于 p_e^c。

一方面，由于短距离链路使用的通常是免费的频带并具有比蜂窝链路高得多的数据速率，因此我们通常认为用于短距离链路上的包交换的网络资源是免费的。另一方面，

考虑到越来越多的应用开始利用成本较低的短距离链路，我们有必要通过一种有效的方式来利用短距离的网络资源。下一节我们将讨论如何以更有效的方式在短距离链路上实现数据包交换。

5.5 网络编码应用于用户协同

将网络编码应用于用户协同的主要动机是提高短距离链路的协同效率。首先，网络编码可以明显减少移动设备之间交换的数据包数量。尤其是使用网络编码时，在协同集群中为了解码所有可恢复的丢包，所需交换的数据包的数量将小于 $|\hat{L}| - |\Lambda|$。因为编码数据包是多个数据包的组合，所以多个丢失了不同数据包的节点可以从相同的编码包中受益。此外，对于部分连接的协同集群，即集群内的节点相互不能直接通信，但信息可以在集群内被中继，网络编码的再编码特性使得包交换效率非常高。因此，网络编码有助于减少整体流量以及用于短距离链路协同交换的能量。下面分析本地交换的编码数据包的数目。

令 i 和 j 分别表示丢包最多和丢包最少的移动设备，即

$$|L_i| = \max\{|L_1|, |L_2|, \cdots, |L_n|\} \tag{5-15}$$

$$|L_j| = \min\{|L_1|, |L_2|, \cdots, |L_n|\} \tag{5-16}$$

当移动设备 i 和 j 没有任何公共丢失包时，移动设备 j 可以发送编码数据包来修复移动设备 i 的所有丢包，反之亦然。这些编码数据包同样可以用来纠正协同集群中的其他移动设备的丢包。即使这些设备的丢包各不相同，只要它们的可恢复丢包的数量小于 i 的可恢复丢包的数量，我们就能够构造一个编码方案，来保证所有其他移动设备都可以通过监听编码数据包来恢复丢包。

当移动设备 i 的丢包得到纠正后，它就可以发送编码数据包来纠正移动设备 j 的丢包。在这种情况下，协同集群中交换的编码数据包的数量 N_{nc} 可以表示为

$$N_{nc} = (|L_i| - |\Lambda|) + (|L_j| - |\Lambda|) \tag{5-17}$$

其中，$|\Lambda|$ 是集群丢失数据包的数目，也就是移动设备无法通过本地重传恢复的那些丢包。

还有一种情况是，除了 Λ 中的数据包以外，移动设备 i 和 j 还有一些公共丢失包，这些丢失包可以由集群中的其他节点来帮助纠正。这种情况下的 N_{nc} 小于前一种情况，即

$$N_{nc} < (|L_i| - |\Lambda|) + (|L_j| - |\Lambda|) \tag{5-18}$$

该式给出了通过短距离链路交换的编码数据包的上界。

下面从另一个角度来看待这个问题。令 Δ_k 表示只有移动设备 k 接收到的数据包的集合，则 Δ_k 满足

$$\Delta_k = \Lambda_{-k} \setminus \Lambda \tag{5-19}$$

其中，$\Lambda_{-k} = L_1 \cap L_2 \cap \cdots L_{k-1} \cap L_{k+1} \cap \cdots \cap L_n$。

移动设备 k 要想解码它所有的可恢复的丢包，它必须接收 $|L_k| - |\Lambda|$ 个数据包。由于 k 也是唯一拥有这些包的设备，故它还必须发送 $|\Delta_k|$ 个数据包。因此，节点 k 发

送和接收的交换数据包的数目 $|\Omega_k|$ 可以表示为

$$|\Omega_k| = (|L_k| - |\Lambda|) + |\Delta_k| \tag{5-20}$$

假设 $|\Omega_k|$ 是集合 $\{|\Omega_1|, |\Omega_2|, \cdots, |\Omega_n|\}$ 中的最大一个。如果 $|L_i| \leqslant \max\{|\Omega_1|, |\Omega_2|, \cdots, |\Omega_n|\}$，则移动设备 k 在该集群中通过网络编码交换了 $|\Omega_k|$ 个编码数据包后，不仅移动设备 i 可以恢复丢失的数据包，而且移动设备 k 独有的 $|\Delta_k|$ 个数据包也将被分布到集群中，因而有

$$N_{nc} = \max\{|\Omega_1|, |\Omega_2|, \cdots, |\Omega_n|\} \tag{5-21}$$

如果 $|L_i| > \max\{|\Omega_1|, |\Omega_2|, \cdots, |\Omega_n|\}$，则集群中交换的数据包须至少为 $|L_i|$，因此有

$$N_{nc} > \max\{|\Omega_1|, |\Omega_2|, \cdots, |\Omega_n|\} \tag{5-22}$$

该式给出了在短距离链路上交换的编码数据包的下界。

故而，我们可以得到 N_{nc} 的范围如下：

$$\max\{|\Omega_1|, |\Omega_2|, \cdots, |\Omega_n|\} \leqslant N_{nc} \leqslant |L_i| + |L_j| - 2|\Lambda| \tag{5-23}$$

N_{nc} 是在短距离链路上交换的数据包的理想数目。为了精确起见，接收到的编码数据包线性独立的概率和短距离链路的误包率均应被纳入考虑范围。

假定发送端拥有 g 个线性独立的符号，则接收到的编码符号与已有的符号均线性独立的概率为[20]

$$P_{ind} = 1 - \frac{1}{q^{g-g'}} \tag{5-24}$$

其中，g' 是汇聚节点接收到的独立符号的数量。

因此要解码一个分代所需接收的编码符号的数量的期望值可由下式计算得到：

$$E_g = \sum_{g'=0}^{g-1} \left(1 - \frac{1}{q^{g-g'}}\right)^{-1} \tag{5-25}$$

故而为了在本地纠正所有可恢复的丢包所需交换的编码数据包的数量可以表示为

$$N_{sr}^{nc} = \frac{N_{nc} E_g}{1 - p_e^{sr}} \tag{5-26}$$

当 q 较大时，$E_g \approx 1$，因此

$$N_{sr}^{nc} \approx \frac{N_{nc}}{1 - p_e^{sr}} \tag{5-27}$$

N_{nc} 的值实际上依赖于基于网络编码的本地重传方案。换句话说，如果本地重传方案设计合理，N_{nc} 将趋于下界。有很多种可能的方案可以实现本地重传，我们给出其中一种作为示例。其基本思想是首先由当前“最好的”移动设备，即包丢失最少的设备，对其接收到的所有数据包进行编码并发送。

该编码数据包能带来两方面的好处：一方面，该编码数据包能够纠正其他移动设备上的大部分丢失的概率是最高的；另一方面，它含蓄地指示出了丢失的数据包。当前“最好的”移动设备丢失的数据包可以看做是当前“罕见的”数据包。

移动设备在发送其编码数据包之前，会根据其退避定时器等待一段确定的时间。定

时器的值是移动设备拥有的数据包的数量和罕见数据包的数量的函数。移动设备拥有的“罕见”数据包越多，定时器的值越小。因而，通过接收编码数据包，一些“非最好”的移动设备变得“更好”，即移动设备丢失的数据包将变少。

接着，这些移动设备中的一个将变成当前“最好的”移动设备，并由当前“最好的”移动设备继续开始发送编码数据包，直到它被另一个“更好的”设备所取代。每当当前“最好的”移动设备发送一个包含该分代中所有数据包的编码包，其他所有设备都将重置一次它们的回退计时器。接着该回退计时器将被用于发送反馈，其值是移动设备所需数据包的数量的函数。移动设备需要的数据包越多，其回退时间就越小。因此，当前“最差的”移动设备可以发送一个简短的反馈来表明它还需要多少数据包，这样，当前“最好的”移动设备就可以在发送完所需的数据包数量后停止发送。

5.6 仿真结果

本节将给出两类主要的仿真结果，首先给出采用用户协同时的本地重传增益，其次给出将网络编码应用于短距离链路获得的增益。

根据 5.2 节的分析可知，纠删性能取决于源块的大小。源块越小，承载的开销将越大。虽然使用较小的源块有利于保证服务质量，但当源块小于 1024B 时，传输效率低下，因此 3GPP 限定源块大小不低于 1024B。利用用户协同，源块大小可以被进一步降低至 512B 甚至 256B。减小源块的大小会降低感知延迟。

假设平均误包率为 10%，图 5-7 给出了对用户协同的仿真结果。当源块大小等于 1024B 且协同集群中有 2、3、4 个移动设备时，蜂窝链路上节省的开销分别为 80.6%、89.0%

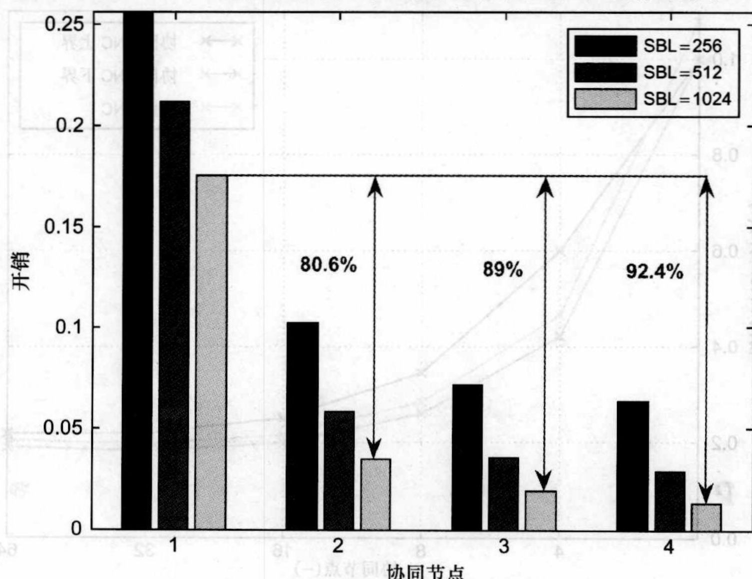


图 5-7　使用用户协同时，蜂窝链路上的开销节省

和92.4%。仿真结果表明，利用用户协同可以显著地减少开销。同时也可以看出，对于大型协同集群，Raptor编码已经不再适合。此外，对于有两个设备的协同集群，当数据块大小等于512B和256B时的开销分别只有6%和10%。该开销甚至要低于不使用协同的情况下当数据块大小等于1024B时的开销。这表明对于两个用户的情况使用较小的源块，如512B或256B是可行的。此外，使用源块大小为512B和256B的源块时，感知延迟分别减少了50%和75%。综上所述，即使是在已有的最先进的系统中，也只能通过牺牲延迟来减少开销，反之亦然。但利用用户协同则可以同时获得低时延和低开销。

为了展示网络编码的主要好处，我们比较了在正常用户协同下和在采用了网络编码的用户协同下恢复所有丢失所需的交换数据包的数目。该仿真的基本假设是：以64个数据包为单位进行本地重传，即网络编码的分代大小为64；蜂窝链路和短距离链路的误包率分别是10%和5%。仿真结果如图5-8和图5-9所示。当协同设备的数量超过两个时，网络编码开始工作。结果表明，基于网络编码的协同所需交换的数据包要少于没有网络编码的协同。换而言之，在所需交换的数据包数量方面，基于网络编码的协同在短距离链路上减少了协同代价。从图5-8中可以看出，当协同集群中有4个设备时，使用启发式网络编码方法可以节省超过一半的交换数据包；当协同集群中有8个设备时，可以节省大约75%的交换数据包。此外还可以看出，由于基于网络编码的协同所需的交换数据包较少，因此与没有网络编码的协同相比，它可以在更短的时间内恢复所有的丢失。图5-9表明，当基于网络编码的协同节点数量增加时，交换数据包的数量将平滑增加，而在不使用网络编码的情况下，它将显著增加。这表明基于网络编码的协同在大协同集群中具有更好的可扩展性。另外，图5-8和图5-9均表明所提出的基于网络编码的本地重传，即启发式网络编码方法，其性能非常接近推导出的性能下界。

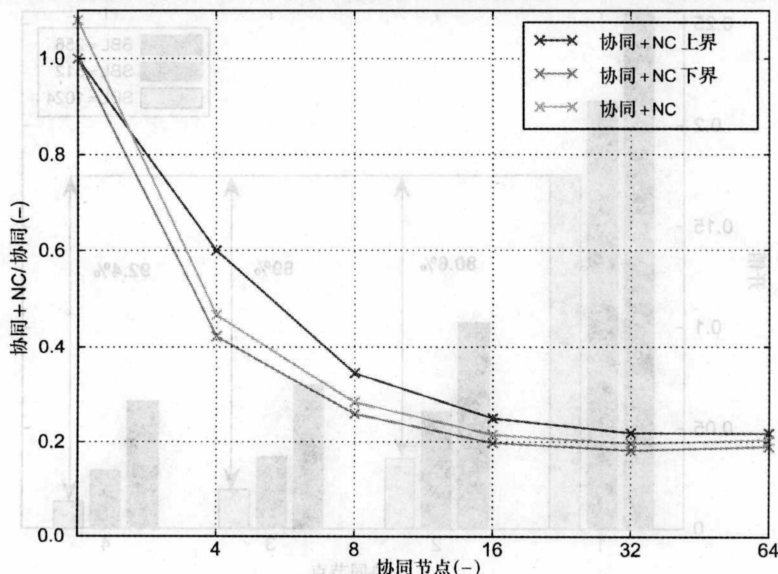


图5-8　使用网络编码时，短距离链路上的协同成本节省

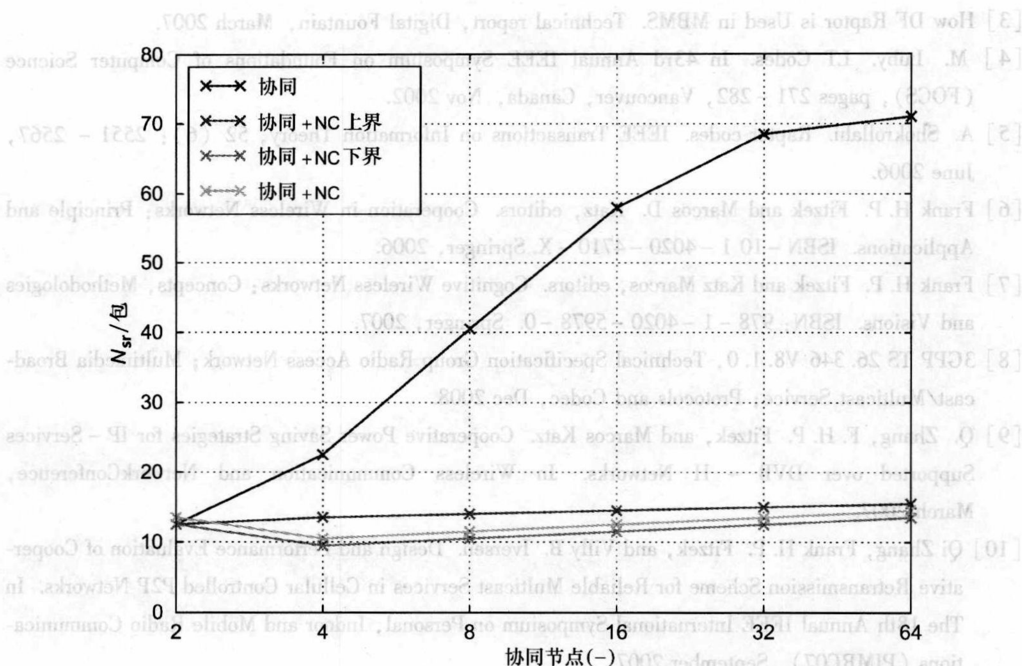


图 5-9　采用/不采用网络编码时，在短距离链路上交换的数据包的数量比较

5.7　结论

为了解决 Raptor 码在 eMBMS 中的弊端，我们在使用 Raptor 码的基础上提出了一种新的基于用户协同概念的本地重传方案。仿真结果表明，只要有两个或多个移动设备进行协同，蜂窝链路上的本地重传就可以节省约 80% 的开销。通过增加协同设备的数目还可以实现更大的增益。此外，本地重传使得在采用 Raptor 码的蜂窝链路上可以使用更小的源块，从而可以降低用户的感知时延，提高用户的感知 QoS。为了使短距离链路上的本地重传效率更高，我们考虑将网络编码应用于本地重传并提出了一个本地重传协议。仿真结果表明，当协同集群中有 4 个移动设备时，网络编码可以节省短距离链路上的超过一半的流量。减少短距离链路上的流量可以减少总体能耗，并减少完成本地数据包交换所需的时间，尤其是对于流量密集的网络。

参考文献

[1] http：//www. 3gpp. org/.

[2] 3GPP TS 26. 346 V6. 6. 0，Technical Specification Group Services and System Aspects；Multimedia Broadcast/Multicast Service（MBMS）；Protocols and codecs，Oct. 2006.

[3] How DF Raptor is Used in MBMS. Technical report, Digital Fountain, March 2007.

[4] M. Luby. LT Codes. In 43rd Annual IEEE Symposium on Foundations of Computer Science (FOCS), pages 271 - 282, Vancouver, Canada, Nov 2002.

[5] A. Shokrollahi. Raptor codes. IEEE Transactions on Information Theory, 52 (6): 2551 - 2567, June 2006.

[6] Frank H. P. Fitzek and Marcos D. Katz, editors. Cooperation in Wireless Networks: Principle and Applications. ISBN - 10 1 - 4020 - 4710 - X. Springer, 2006.

[7] Frank H. P. Fitzek and Katz Marcos, editors. Cognitive Wireless Networks: Concepts, Methodologies and Visions. ISBN: 978 - 1 - 4020 - 5978 - 0. Springer, 2007.

[8] 3GPP TS 26. 346 V8. 1. 0, Technical Specification Group Radio Access Network; Multimedia Broadcast/Multicast Service; Protocols and Codec, Dec 2008.

[9] Q. Zhang, F. H. P. Fitzek, and Marcos Katz. Cooperative Power Saving Strategies for IP - Services Supported over DVB - H Networks. In Wireless Communication and NetworkConference, March 2007.

[10] Qi Zhang, Frank H. P. Fitzek, and Villy B. Iversen. Design and Performance Evaluation of Cooperative Retransmission Scheme for Reliable Multicast Services in Cellular Controlled P2P Networks. In The 18th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC07), September 2007.

[11] Qi Zhang, Frank H. P. Fitzek, and V. B. Iversen. One4all Cooperative Media Access Strategy in Infrastructure Based Distributed Wireless Networks. In IEEE Wireless Communication and Networking Conference (WCNC), April 2008.

[12] M. Luby, M. Watson, T. Gasiba, and T. Stockhammer. Mobile Data Broadcasting over MBMS Tradeoffs in Forward Error Correction. In the 5th International Conference on Mobile and Ubiquitous Multimedia (MUM), Stanford, CA, USA, December 2006.

[13] P. Fuxjager and F. Ricciato. Collecting Broken Frames: Error Statistics in IEEE 802. 11b/g Links. In Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks andWorkshops, 2008. WiOPT 2008. 6th International Symposium on, pages 30 - 35, April 2008.

[14] Janus Heide, Morten V. Pedersen, Frank H. P. Fitzek, Tatiana V. Kozlova, and Torben Larsen. Know Your Neighbour: Packet Loss Correlation In IEEE 802. 11b/g Multicast. In 4th International Mobile Multimedia Communications Conference (MobiMedia), Oulu, Finland, 7 - 9 July 2008.

[15] Hoi - Sheung Wilson So, Kevin Fall, and Jean Walrand. Packet Loss Behavior in a Wireless Broadcast Sensor Network. http: //citeseerx. ist. psu. edu/.

[16] M. Luby, A. Shokrollahi, M. Watson, and T. Stockhammer. Raptor Forward Error Correction Scheme for Object Delivery, RFC 5053, October 2007.

[17] F. H. P. Fitzek, M. Katz, and Q. Zhang. Cellular Controlled Short - Range Communication for Cooperative P2P Networking. In Wireless World Research Forum (WWRF) 17, volume WG 5, Heidelberg, Germany, November 2006. WWRF.

[18] F. H. P. Fitzek, M. Katz, and Q. Zhang. Cellular Controlled Shortrange Communication for Cooperative P2P networking. Wireless Personal Communications, 2008.

[19] T. K. Madsen, Q. Zhang, and F. H. P. Fitzek. Design and Evaluation of IP Header Compression for Cellular - Controlled P2P Networks. In IEEE International Conference on Communication (ICC), June 2007.

[20] A. Eryilmaz, A. Ozdaglar, and M. Medard. On Delay Performance Gains From Network Coding. Information Sciences and Systems, 2006 40th Annual Conference on, pages 864 - 870, March 2006.

第6章　CONCERTO：现实世界 MANET 系统中的一些基于网络编码的经验

Victor Firoiu 美国，马萨诸塞州，伯林顿，BAE 系统，先进信息技术公司

Greg Lauer 美国，马萨诸塞州，剑桥，BBN 技术公司

Brian Decleene 美国，马萨诸塞州，伯林顿，BAE 系统，先进信息技术公司

Soumendra Nanda 美国，马萨诸塞州，伯林顿，BAE 系统，先进信息技术公司

摘要

虽然网络编码在提高系统性能方面带来了许多潜在的好处，但是其大部分实践经验都仅限于模拟、仿真和小规模演示。本章将介绍一个已经完全实现的基于网络编码协议栈的系统：CONCERTO。我们在基于当前主流的 802.11 协议集的移动自组织网络（MANET）中实地测试并比较了 CONCERTO 和一组基准协议集，我们将在本章中回顾这些测试结果。这些实地测试是基于一些配备有移动无线电台（行人、车辆和空中中继节点）的操作场景，并使用视频、文件传输、聊天及各种环境感知应用进行搜索和救援任务。相比于目前最先进的协议集，CONCERTO 系统能带来 2 ~ 3 倍的视频吞吐量提升，以及高达 7 倍的距离 - 效用乘积提升。本章将探讨关于嵌入式系统中的网络编码的一些实际实现问题，并分析其性能结果。

关键词

网络编码、移动 Ad Hoc 网络、无线网络、路由、可靠多播、网络协议。

6.1　引言

移动自组织（Ad Hoc）网络（MANET）已成为野外作业中军事通信和第一响应通信的一个重要组成部分。尽管无线电技术在逐渐发展，商用无线通信也在逐渐普及，然而 MANET 提供的容量仍然仅占其潜在容量的极小一部分。其原因有：频谱资源的稀缺，稳定基础设施的匮乏，不能容忍“忙”信号的非商业需求，比如关键任务系统，以及针对具有本质区别的“有线”情景设计的基于 IP 的分层技术的使用。

基于控制的移动 Ad Hoc 联网（CBMANET）DARPA 项目的目标是：研究、设计、开发出一个革命性的移动 Ad Hoc 网络原型并对其进行测试，相比当前的网络发展水平，该原型应更能从网络利益相关者（用户和运营商）的角度提高有效性性能。

BAE 系统团队开发了一个 MANET 解决方案，被称为基于网络编码的增强型无线传输优化中的网络编码控制系统（CONCERTO）。基于网络编码提出的总体方案允许网络运营商充分实现无线多播的理论吞吐能力；可以高效地解决多播优化问题，在使用其他方法时该问题是一个 NP 完全问题；提供了对于损耗和路由环路的鲁棒性；统一了单

播、多播、广播和多路径算法；在网络链路上实现了无速率编码；利用了机会传输。通过一系列现场测试，我们证明了MANET通信系统的可靠性和鲁棒性，并展示了在实际移动无线场景中获得的显著性能提升。

6.1.1 无线MANET面临的挑战

设计无线Ad Hoc网络中的高效可靠的通信方案面临着一些挑战，尤其是当网络中使用的是针对有线网络提出的基于IP的协议集时。首先，MANET的拓扑是高度动态的，网络中的连接以秒级或者更短的时间为单位变化着。所有基于最短路径策略的IP路由都会变得失效并需要被重新计算，这将导致传输中断以及路由控制开销增加。

另一个挑战是，由于物理层多径之间的自干扰、不同设备之间的互干扰和/或网络拥塞，无线链路是易受损的。如此高且多变的链路损耗给不区分链路损耗和拥塞损耗的现有的可靠传输协议带来了极大的挑战，这些协议包括TCP协议和基于否认的可靠组播（NORM）协议[1]等。

高路损还与包括TCP和NORM在内的现有的传输协议中的端到端原则相悖。根据该原则，我们仅在端节点（源节点和目的节点）进行可靠性控制（检查丢失的数据并重发）。这将导致传输效率随着路径总跳数的增加而指数下降（因为端到端成功传输的概率等于每段链路成功传输概率的乘积）。

网络中过多的节点数量对于MANET来说也是一个巨大的挑战。虽然现有的一些MANET协议，如简单多播转发（SMF）[2]等，在小规模网络（10～15个节点）中能够克服上述问题，但是当网络中的节点数量大于20的时候，它们会变得低效。实际上，虽然在整个网络中洪泛每个数据流的策略可以减轻路由计算复杂和链路不可靠的问题，但是对于目的节点仅占整个网络中节点的一小部分的一般的多播场景来说，洪泛策略是十分低效的。此外，从单播（一个目的节点）到多播（多个目的节点）再到广播（网络中的所有节点都是目的节点），应用程序的需求范围十分广泛，但即使是目前最先进的协议也仍需要对每种情况单独提供迥然不同的解决方案。

目前所有的解决方案都是基于现今的在“有线”假设下提出的IP分层技术。其分层原则——在网络协议层（链路/MAC层、路由层和传输层）之间隐藏信息——在简化有线网络中的协议设计和操作方面取得了很大的成功。而在无线MANET中正好相反，网络协议可以利用低层信息来克服拓扑结构、状态和无线数据接收的不确定性，从而充分挖掘网络性能的潜力。

6.1.2 CONCERTO解决方案

CONCERTO系统是一个路由层和传输层的联合协议，它允许一个源节点与一组目的节点通信，并支持在单个框架下实现全范围应用（单播、多播和广播）。

CONCERTO的核心是两个可以解决上述无线MANET所面临的挑战的全新思想。第一个思想是以网络编码数据的形式传输信息，以此取代传统的数据包转发。通过对来自相同会话（相同源节点）的数据包进行随机线性组合，我们可以把应用层数据转化成

编码数据包，也可以发送、接收并有可能会在中间节点再次组合这些编码数据包。目的节点不需要考虑所接收到的包的身份标识，当它接收到足够数量的编码包时，它就可以解码得到应用层数据（这是网络编码理论的一个基本结论，详细内容见第6.3节）。因此，网络编码将数据传输从传统的特定数据包转发方式（需要识别传输的是哪个包，检查是否丢包，如果丢包则需重传）转变成了只关注接收到的编码包的数量而不需要关注接收到的是哪几个包的信息传播方式。6.5节中描述的网络编码传输模块取代了诸如TCP和NORM这样的传输层协议，并与6.4节中描述的子图构造模块紧密耦合。

CONCERTO系统的第二个核心思想是基于子图的数据传播。一个子图指的是将一个会话信息从其源节点转发到其（若干个）目的节点的过程中使用到的所有节点和链路的集合。IP单播（沿最短路径转发）和IP多播（沿多播树转发）可以看做是该思想的特殊情况。子图的构造是基于求解描述源节点和目的节点之间的网络编码信息流的方程。子图转发能够简单地解决路由环路的检测和消除问题，实现可靠的多播传输，并利用机会接收（附近的节点均可接收，而不是只有指定的“下一跳”才能接收）——这些都是在传统的IP框架下难以解决的问题。最重要的是，CONCERTO仅允许全部节点的一个子集发送某个会话的数据包，这使得CONCERTO系统的效率远远超过洪泛策略。6.4节中描述的子图构造器可以为给定的多播会话选择有效的子图。该模块可以取代传统的IP最短路径路由算法（用于开放式最短路径优先（OSPF）[3]以及开放式链路状态路由（OLSR）[4]等协议）或洪泛协议（如简单多播转发（SMF）[2]）。

6.2 CONCERTO综述

CONCERTO系统的主要组成如下（见图6-1）：

- 应用程序接口（AppIF）：从通过以太网与CONCERTO相连的PC上运行的应用程序中接收数据包，将它们按照传输类型分类，并把它们交给网络编码器。
- 网络编码器（NetCoder）：以分代为单位划分应用层数据包，对同一分代内的数据包进行随机线性编码，根据需求给转发引擎提供编码包，并在目的节点处对数据包解码。我们将在6.3节中详细讨论该模块。
- 子图构造器（SG Constructor）：利用关于网络拓扑结构、多播组、源和目的节点的信息，为每个应用层会话选择转发该会话的网络编码包的节点，并为每个节点确定其需转发的流量。我们将在6.4节中详细讨论该模块。
- 转发引擎（Master Fwder，也表示为Fwdr）：采用新的传输协议来判断分别需要对哪些数据包进行编码、转发、解码、丢弃、重传、超时操作。这些将在6.5节中进行详细描述。
- 邻居发现，拓扑发现（拓扑缓存，也称为Topo缓存）：确定节点的连接性并估计信道质量；使用高效的最优链路状态路由协议（OLSR）在整个网络中传播连接信息，以支持子图的构造和多播组成员的分配。
- 组管理（因特网组管理协议（IGMP）模块，也称为组缓存）：跟踪记录每个多

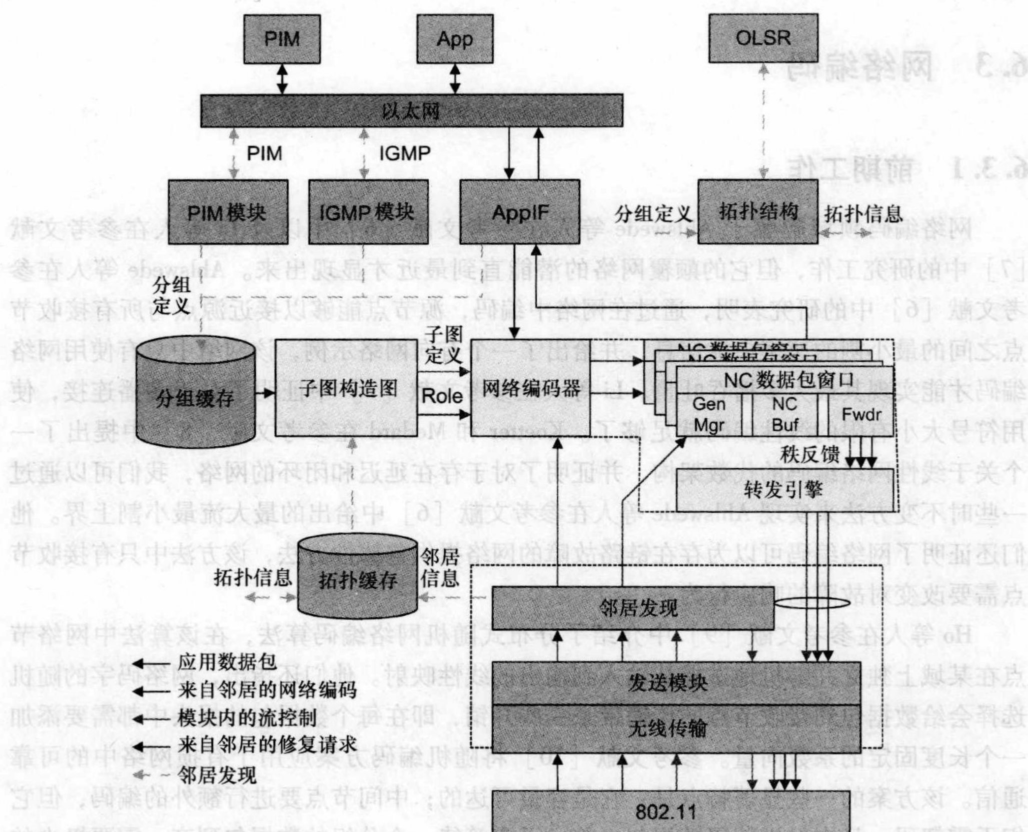


图 6-1　CONCERTO 架构

黑线表示对该节点应用层产生的数据包的处理流程；蓝线表示对接收到的来自邻居节点的网络编码包的处理流程（转发/解码）；红线表示来自邻居节点的恢复请求数据包；橙线表示周期性的邻居发现数据包；绿色的虚线表示模块内的控制流。黄色的模块与具体的网络编码相关，本章的其余部分将对它们进行详细的讨论

播组都有哪些目的节点，并把该信息提供给子图构造器。

- 网关功能（协议独立多播（PIM）模块）：在基于 IP 的 PIM 多播路由[5] 和基于 CONCERTO 的网络之间提供互操作性。

本章的其余部分结构如下：

6.3 节简要介绍了网络编码的基本原理以及如何在 CONCERTO 中应用网络编码。6.4 节概述了用于选择参与转发某会话的网络编码包的节点，并确定其参与程度的算法和模块。6.5 节描述了用于转发某会话的网络编码包的协议。6.6 节描述了除了理论吞吐量提升，网络编码带来的其他效益。6.7 节介绍了用于实现并评估基准方法和 CONCERTO 算法的基于仿真的解决方案。6.8 节讨论了 CONCERTO 算法在一系列现场实验中表现出的性能，并就为什么 CONCERTO 算法比基准算法的性能好这个问题给出了深刻的见解。6.9 节讨论了关于如何进一步提高性能的一些见解。

6.3 网络编码

6.3.1 前期工作

网络编码领域起源于 Ahlswede 等人在参考文献［6］中以及 Li 等人在参考文献［7］中的研究工作，但它的颠覆网络的潜能直到最近才显现出来。Ahlswede 等人在参考文献［6］中的研究表明，通过在网络中编码，源节点能够以接近源点与所有接收节点之间的最小割的速率多播信息，并给出了一个有向网络示例，该网络中只有使用网络编码才能实现其最大多播吞吐量。Li 等人在参考文献［7］中证明了对于多播连接，使用符号大小有限的线性编码就足够了。Koetter 和 Medard 在参考文献［8］中提出了一个关于线性网络编码的代数架构，并证明了对于存在延迟和闭环的网络，我们可以通过一些时不变方法来实现 Ahlswede 等人在参考文献［6］中给出的最大流最小割上界。他们还证明了网络编码可以为存在链路故障的网络提供鲁棒性方法，该方法中只有接收节点需要改变对故障的响应行为。

Ho 等人在参考文献［9］中介绍了分布式随机网络编码算法，在该算法中网络节点在某域上独立并随机地选择从输入到输出的线性映射。他们还指出，网络码字的随机选择会给数据包到接收节点的传输带来一些开销，即在每个数据包的报头中都需要添加一个长度固定的系数向量。参考文献［10］将随机编码方案应用于有损网络中的可靠通信。该方案的一些显著特点是：它是容量可达的；中间节点要进行额外的编码，但它们无需解码，且在发送编码数据包之前也无需等待一个分组的数据包到齐；需要极少的甚至不需要反馈或协调控制；所有的编码和解码操作都可以在多项式时间内完成。

6.3.2 CONCERTO 网络编码

CONCERTO 采用的是基于分代的随机分布式网络编码（见图 6-2）。对于一个给定的编码子图（参与给定数据会话的网络编码传输的网络节点的集合），参考文献［9］证明了该方法能够实现子图中的单个多播连接的容量。因此，在网络中建立一个最优的多播会话时，对子图选择问题和编码问题进行拆分不会影响其最优性，也就是说我们可以把该最优化问题拆分为将在 6.4 节中详细讨论的最小代价子图选择问题，以及接下来要讨论的在给定子图中的码字构造问题。

假设源节点 S 将一连串数据包按每 g 个划分为一个分代，第 i 个分代由数据包 $w_1^i(s)$，$w_2^i(s)$，…，$w_g^i(s)$ 组成。下面考虑节点 n，假设 n 已经接收到了第 i 代中的 k 个信息 $w_1^i(s)$，$w_2^i(s)$，…，$w_k^i(s)$。每个信息都可以看做是有限域$\mathbb{F}_q$上的一个长度为 ρ 的向量（假设数据包的长度是 b 比特，则有 $\rho = \lceil b/\log_2 q \rceil$）。所有节点都进行相同的编码操作：对接收到的数据包求线性组合，从而产生新的待传输的数据包，其线性组合系数随机选取自域$\mathbb{F}_q$。由于所有的编码都是线性的，因此网络中的任意数据包 x 都可以表

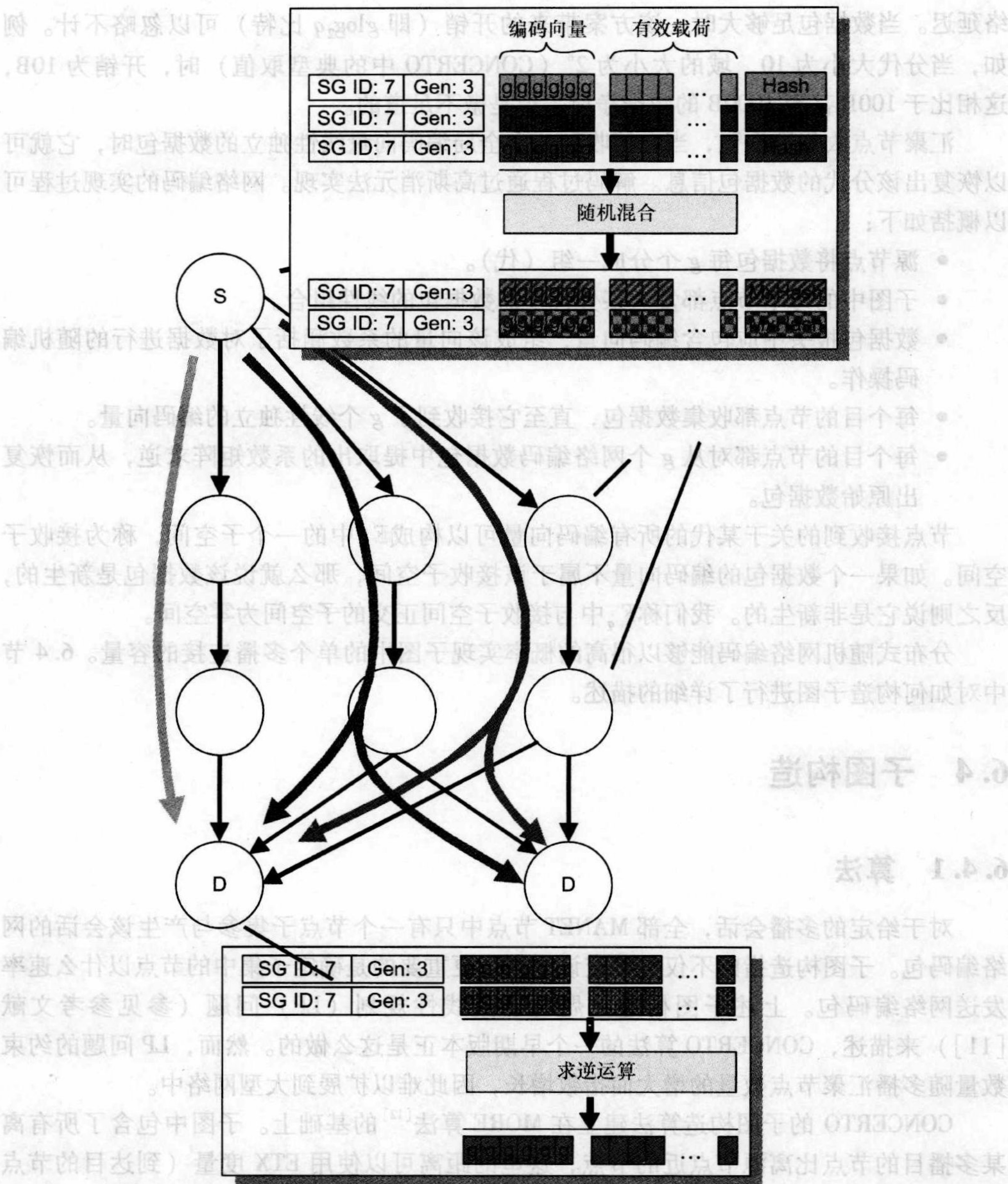


图6-2 每个网络编码包中都有一个大小固定的向量，该向量指示了与源数据包对应的线性编码系数。目的节点将使用这些系数解码该代数据包

示成 $w_1^i(s)$，$w_2^i(s)$，…，$w_g^i(s)$ 的线性组合，即 $x=\sum_{j=1}^{g}\gamma_j w_j^i(s)$，其中 γ_j 是 $\mathbb{F}_q$ 中的元素。我们称 $\gamma=(\gamma_1, \gamma_2, \cdots, \gamma_g)$ 为 x 的全局编码系数，并假设该系数将被放在 x 的报头中随 x 一起发送。注意中间节点不是非要收够 g 个数据包才能发送，这会降低网

络延迟。当数据包足够大时，该方案带来的开销（即 $g\log_2 q$ 比特）可以忽略不计。例如，当分代大小为 10，域的大小为 2^8（CONCERTO 中的典型取值）时，开销为 10B，这相比于 100B 甚至 1000B 的数据净荷来说是微不足道的。

汇聚节点收集数据包，当它接收到 g 个全局编码向量线性独立的数据包时，它就可以恢复出该分代的数据包信息。解码过程通过高斯消元法实现。网络编码的实现过程可以概括如下：

- 源节点将数据包每 g 个分成一组（代）。
- 子图中的每个节点都发送某分代中的数据包的线性组合。
- 数据包报头中应包含编码向量，组成该向量的系数涵括了对数据进行的随机编码操作。
- 每个目的节点都收集数据包，直至它接收到了 g 个线性独立的编码向量。
- 每个目的节点都对从 g 个网络编码数据包中提取出的系数矩阵求逆，从而恢复出原始数据包。

节点接收到的关于某代的所有编码向量可以构成$\mathbb{F}_q^g$中的一个子空间，称为接收子空间。如果一个数据包的编码向量不属于该接收子空间，那么就说该数据包是新生的，反之则说它是非新生的。我们称$\mathbb{F}_q$中与接收子空间正交的子空间为零空间。

分布式随机网络编码能够以很高的概率实现子图中的单个多播连接的容量。6.4 节中对如何构造子图进行了详细的描述。

6.4 子图构造

6.4.1 算法

对于给定的多播会话，全部 MANET 节点中只有一个节点子集参与产生该会话的网络编码包。子图构造指的不仅是确定该子集，更重要的是确定子集中的节点以什么速率发送网络编码包。上述子图构造问题可以用线性规划（LP）问题（参见参考文献[11]）来描述，CONCERTO 算法的一个早期版本正是这么做的。然而，LP 问题的约束数量随多播汇聚节点数量的增大而指数增长，因此难以扩展到大型网络中。

CONCERTO 的子图构造算法建立在 MORE 算法[12]的基础上。子图中包含了所有离某多播目的节点比离源节点近的节点，这里的距离可以使用 ETX 度量（到达目的节点所需的期望传输次数）来测量。每当子图中的节点接收到一个新生的数据包时，它就会产生并发送一个网络编码包的转发因子值。源节点负责计算子图中的每个节点的转发因子，并把它放在每个数据包的报头里注入子图中。每当拓扑结构或是信道质量发生改变时，允许源节点重新计算转发因子。为了减少信道接入竞争，转发因子低于某一阈值的节点将不被包含在子图中。

虽然 CONCERTO 的子图和转发因子都是通过 MORE 算法计算得到的，但 CONCERTO 原创了可靠传输机制来确定什么时候需要额外的传输，什么时候可以停止发送网络

编码包。这些机制将在 6.5 节中讨论。

6.4.2 实现

子图构造器与 CONCERTO 系统其他部分的交互如图 6-3 所示。

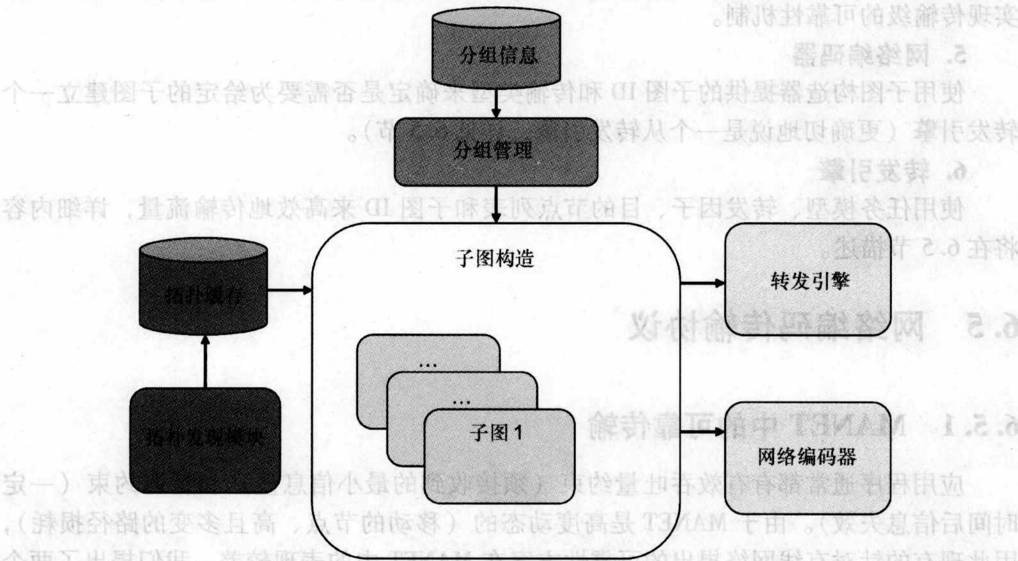


图 6-3　子图构造器的系统示意图

1. 组管理

每个多播会话都单独使用一个多播 IP 地址。组管理采用 IGMP 协议[13]来收集关于每个多播组都有哪些节点的信息。并使用 OLSR 协议[4]洪泛以下信息：

- 组 ID；
- 源节点；
- 目的节点；
- 应用类型：弹性全可靠（文件传输），非弹性半可靠（流媒体、聊天、态势感知 – SA）㊀。

2. 拓扑发现模块

拓扑发现模块利用来自邻居节点的信标和监听流量动态地发现节点并确定链路质量。它将这些信息存入拓扑缓存。远程链路的质量信息由 OLSR 的链路状态传播协议提供。除此之外，CONCERTO 不再使用 OLSR 中的其他函数（如最短路径路由等）。

3. 拓扑缓存（TopoCache）

拓扑缓存是一个对象数据库，它包含了网络中所有的节点、连接这些节点的链路以及相关的链路质量信息。

㊀ 简单起见，不同类型的服务使用不同范围的 IP 地址。

4. 子图构造器

子图构造器对组记录进行迭代并计算子图信息。它使用到的信息来自拓扑缓存（存储了关于网络中的节点、链路以及每条链路的丢失率的信息）和组管理（把组播地址映射为服务类型需求）单元。网络编码器和转发引擎使用子图构造器产生的信息来实现传输级的可靠性机制。

5. 网络编码器

使用子图构造器提供的子图 ID 和传输类型来确定是否需要为给定的子图建立一个转发引擎（更确切地说是一个从转发引擎，详见 6.5 节）。

6. 转发引擎

使用任务模型、转发因子、目的节点列表和子图 ID 来高效地传输流量，详细内容将在 6.5 节描述。

6.5　网络编码传输协议

6.5.1　MANET 中的可靠传输

应用程序通常都有有效吞吐量约束（须接收到的最小信息量）和延迟约束（一定时间后信息失效）。由于 MANET 是高度动态的（移动的节点、高且多变的路径损耗），因此现有的针对有线网络提出的可靠性方案在 MANET 中的表现较差。我们提出了两个适合于不可靠 MANET 的可靠传输协议。这些协议充分利用了网络编码的特点，并在有效吞吐量和延迟之间提供了不同的权衡。

- 半可靠：诸如交互式语音和视频流等的应用能够容忍小部分数据包的丢失，但这些应用规定只有在给定时间内投递成功的数据包才是有效的。此外，这类应用可以在一段时间的高损耗后恢复。我们提出的半可靠算法采用逐跳机制和积极的重传策略来确保所有接收节点在可配置的延迟范围内（通常是数百毫秒到几秒的时间）都能够接收到大部分数据包。
- 全可靠：诸如文件传输和电子邮件等的应用要求构成一个数据元素的所有数据包都必须被成功投递——丢失任何一个数据包都会导致整个传输无效。这类应用对延迟的要求通常比交互式应用要宽松。我们提出的全可靠算法在逐跳机制的基础上采用了端到端机制，其目的是在存在网络分区、严重丢包等事件的网络中，降低目的节点不能接收到完整的数据元素的概率。

下文我们将描述协议架构，该架构定义了子图中产生和发送网络编码包的反馈回路、确定下一次调度哪个子图的数据包的主/从架构以及半可靠和全可靠传输协议中的从转发算法。这些算法中有一部分曾受到参考文献［14］中提出的基于网络编码的多播协议 CodeCast 的启发。

6.5.2　转发协议架构

CONCERTO 建立了一个网络编码多播会话，该会话中源节点通过若干节点组成的

一个有向子图与接收节点相连接。在这个有向子图中，每个节点都有下游邻居节点（不包括没有转发功能的目的节点）和上游邻居节点（不包括源节点）。源节点首先把来自应用层的数据包以代为单位分组，然后把每个分代中的数据包的随机线性组合广播给它的下游邻居节点。它的所有下游节点都接收网络编码包，并组合同一分代的编码包，之后把新得到的随机线性组合广播给自己在子图中的下游节点。当目的节点接收到足够多的同一分代的线性独立的编码数据包时，它就可以解码得到原始数据包。

我们提出的算法通过本地修复策略来实时且高效地修复丢损。与 TCP 中的端到端算法不同，本地修复策略使用暂存在中间节点的（网络编码所必需的）数据包来产生新的网络编码包，而无需端到端的反馈。节点的任意上游邻居节点都可以给该节点发送本地修复数据包，这将提高修复本地丢损的概率。

下面通过一个示例网络来说明这个协议，网络中有一个从发送节点 S 到目的节点 D 和 E 的多播会话（见图 6-4）。该会话的子图中有 A、B 和 C 共 3 个转发节点。发送节点把接收到的来自应用层的数据包以代为单位分组，然后产生网络编码包，并通过无线信道以本地广播的方式发送出去（见图中的实线箭头）。

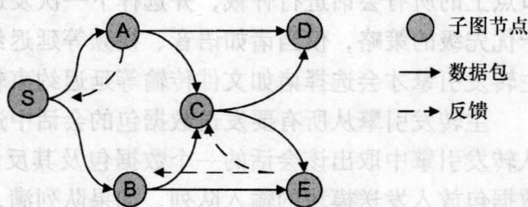


图 6-4　一个由 6 个节点组成的网络，图中标出了数据流和可靠性控制信息流

节点发送的编码包的数量取决于子图构造器计算出来的转发因子。需要注意的是，转发因子补偿的是无线信道的平均丢损量，而不是某分代数据包所经历的实际丢损量。

如果节点在给定时间内接收到的某分代的编码包过少，它就会产生一个反馈请求来表示它需要更多的该分代的新生数据包（见图中的虚线箭头）。为了避免上游节点发送非新生的数据包，我们会在反馈请求中写入关于该节点的零空间的信息（该零空间中的任意数据包在当前时刻对于该节点来说都是新生的）。只要有需要，反馈请求可以被夹带在任一待发送的数据包的报头中，因为该节点的上游邻居节点通常都可以监听到该节点面向下游邻居节点的广播。如果在每个发送的数据包中都夹带完整的零空间信息，则会带来很大的开销，因此我们只把零空间中的一个随机向量写入反馈请求。

上游节点接收到反馈请求后将会检查反馈请求中的零空间随机向量是否属于该节点的接收子空间。如果属于，则上游节点会发送一个该分代数据包的随机线性组合来响应这个请求。如果不属于，则上游节点会忽略这个反馈请求，因为它发送的数据包很可能是非新生的。为了提高效率，节点响应反馈请求的发送强度等于该节点的本地转发因子，如前文所述。我们允许将发送请求冗余地写入多个连续的数据包中，因此即使对于丢包率很高的情况，反馈信息也能以较高的概率送达。

最后我们要注意的是，该算法仅用于确定发送哪些数据包和哪些反馈信息，而这些信息的传输时间将由速率控制（QoS）模块来确定。

6.5.3　网络编码传输协议的主/从架构

在一个节点上可以同时存在来自不同子图或来自同一子图中的不同分代的数据包。

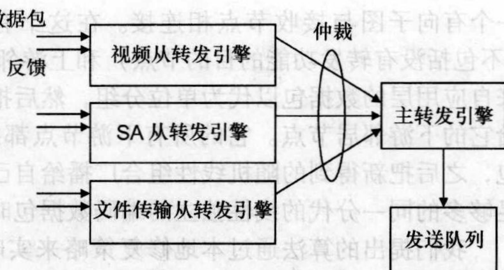


图 6-5　主/从转发架构

图 6-5 中所示的主/从架构用于确定发送哪个子图中的哪个分代的数据包。同时它还能确定是否需要在发送的数据包中夹带修复请求，或者是否需要单独发送一个修复请求。具体而言，对于每个子图，每个节点都有一个从转发引擎，该引擎用于接收该子图的数据和控制信息（反馈）并选择下一次要发送的分代（数据和/或控制信息）。主转发引擎对节点上的所有会话进行仲裁，并选择下一次发送的从转发引擎。主转发引擎采用的是基于优先级的策略，仅当诸如语音、视频等延迟约束较严格的会话没有数据包要发送时，主转发引擎才会选择诸如文件传输等延迟约束较宽松的会话进行发送。

主转发引擎从所有要发送数据包的会话中选择一个优先级最高的会话，并从相应的从转发引擎中取出该会话的一个数据包及其反馈信息（见图 6-6）。然后它将尝试把该数据包放入发送模块的输入队列，如果队列满，则等待直至有数据包被发送使得队列空间可用。如果没有要发送数据包的会话，则主转发引擎将休眠。当任意会话接收到一个新生的数据包，或者定时器表明需要重传某会话时，主转发引擎将被重新激活。

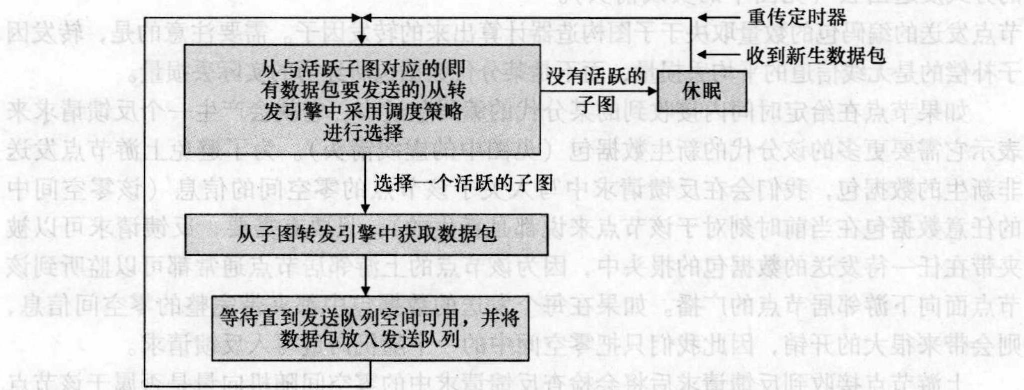


图 6-6　主转发引擎用于选择下一个发送数据包的算法

6.5.4　半可靠从转发引擎算法

半可靠传输算法的设计目的是：

- 最小化每单位应用数据所需传输的数据包数量。
- 最小化将一整代数据从源节点传输到目的节点所需的时间，从而最小化应用的端到端延迟和网络中的分代数量。
- 最小化数据包从创建到传输的延迟，从而使发送数据包中的信息对于下游节点来说是及时且有用的。

半可靠从转发引擎从最早需要发送数据包的分代中产生一个网络编码包，并将它和反馈请求一起组装成下一个发送数据包，反馈请求用于指示所有需要从上游节点获得更多数据包的分代的集合。分代的先进先出的策略可以最小化传输一整代数据所需的时间，进而最小化端到端的传输延迟。

在构造子图时，我们会计算在从源节点到目的节点的通信过程中子图中的每个节点所使用的转发因子。该转发因子的计算通过信用计数机制实现。信用计数器按接收到的新生数据包的转发因子值递增，并按发送出去的数据包数量递减，但仅限于非负范围。如果信用值小于1，则该数据包以等于该信用值的概率进行转发。对于源节点来说，它首先把来自应用层的数据组装成一个完整的分代，然后把信用计数器的信用值初始化为分代大小与转发因子的乘积。

除去由来自上游的新生数据包引起的数据包发送以外，我们可能还需要发送更多的数据包，这是因为无线信道中的丢包量可能会超出转发因子提供给信道的冗余量。当节点确定它没有足够的信息（即节点的某个分代没有满秩）时，它会通过一个反馈信息（如果可能的话，该信息会被夹带在数据中传输）来请求上游节点发送本地修复数据包。如果在一段可配置的时间内节点没有收到新生的数据包，那么它将产生一个上述修复请求。虽然基于反馈的传输效率较高（只发送需要的数据包），但是它也确实会引入延迟。我们提出的算法可以通过调整参数来实现延迟和效率之间的权衡。

考虑到节点有可能会在很短的时间内收到来自许多节点的修复请求，而一个数据包通常可以满足多个请求（众所周知，这是网络编码的一个类似前向纠错（FEC）的性质），因此节点可能会发送冗余的修复数据包。为了最大限度地提高效率，本算法使用了一个定时器来积攒这类请求，并对定时范围内的所有请求响应一个修复数据包。

6.5.5　全可靠从转发引擎算法

诸如文件传输、电子邮件等的应用程序要求目的节点接收到一个文件（或其他数据单元）中的所有数据包，这样这些数据才有价值。大部分这类应用程序的延迟约束都相对宽松，因此可以使用端到端的可靠性协议。

前文描述的半可靠转发算法基于对未满秩分代的检测给出了多种丢包情况下的恢复方法。然而，该算法没有判断节点在加入该子图之前是否已经丢失了分代数据（因为对于延迟敏感型的应用来说，重传最早的分代是没有用的），也没有实现在某些特殊情况下（例如节点没有收到任何当前分代的数据）的处理机制。

全可靠协议在逐跳的半可靠协议的基础上进行了扩展，添加了一些检测并恢复分代丢失的方法。源节点使用试探法来检测文件传输的开头和结尾，并向网络编码数据包流中注入相应的标记。当转发节点或目的节点检测到分代丢失时，它们会发送修复请求。这些请求会向上游传播直到到达某个能够提供该分代的数据包的节点（可能是源节点）。

如果某个目的节点在文件传输的中途加入该多播会话，则源节点可能要重传大量的数据——甚至可能会超出在最宽松的文件传输延迟约束内能够传输的数据量。为了处理

这种情况，我们可以配置全可靠转发引擎，使其能够限制分代的保存时间。

6.6 网络编码的效益

网络编码有一个十分强大的功能，就是它能够广泛容纳其他诸多的 MANET 机制。这意味着网络编码提供了一个可以统一不同网络算法的基础结构——从而大大地简化了这些算法的配置和相互协调。

6.6.1 广播、多播和单播的统一机制

CONCERTO 在一个源节点和一组目的节点之间构造网络编码子图。由于该目的节点组可以只包含一个目的节点，也可以包含网络中除源节点外的所有其他节点，还可以是网络中所有其他节点的集合的一个子集，因此网络编码为单播、多播和广播的实现提供了一个统一的机制。图 6-7 举例说明了一个将信息从源节点 S 传输给目的节点 D_1 和 D_2 的子图。

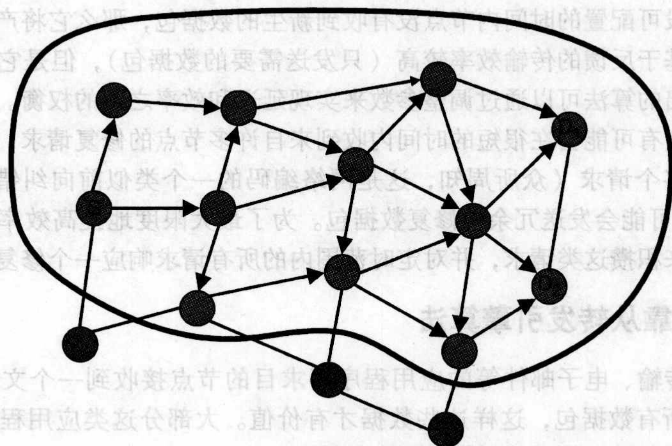


图 6-7　在制定网络编码方案时，我们把单播和广播看做是多播的特殊情况。图中淡蓝色的节点是子图的一部分，它们负责把信息从源节点 S 转发给目的节点 D_1 和 D_2

6.6.2 对于路由环路的鲁棒性

在 MANET 中，拓扑数据传播延迟意味着节点可能没有准确地掌握当前的网络拓扑结构图。在分组转发网络中，这可能会导致转发环路问题。例如，考虑图 6-8 中的场景，网络中存在一条红色的转发环路。当使用（传统的）路由技术时，转发环路会带来严重的问题——可能会导致目的节点不可达，而且由于对同一个数据包无效地进行多次重传，还会导致带宽降低。由于网络编码方案在子图上转发的不是数据包而是信息流（同一分代中的不同数据包的混合），因此采用网络编码技术时，网络中不存在“转发环路”的问题：如果节点接收到的数据包是非新生的，那么该数据包将不会被继续传输。

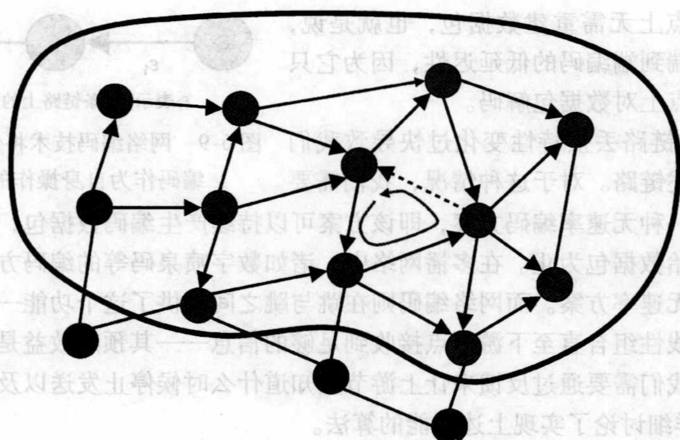


图6-8 由于只有新生的（包含新信息的）数据包能够触发节点的发送行为，因此网络编码对于路由“环路”具有鲁棒性

6.6.3 对于链路或节点故障的鲁棒性

一般来说，最优的子图方案所包含的节点的数量要多于路由网络的生成树所使用到的节点的数量，因此基于子图的传输通常对链路或节点故障具有鲁棒性。在路由网络中，仅通过一条路径转发流量是不可靠的（例如某个节点或者某段链路的丢失会导致目的节点不可达）。如果源节点能够计算出多条源点和目的节点之间的链路不相交或节点不相交路径并使用它们，这种不可靠性就会被减弱。然而，这种方法较为复杂，而且不易被扩展到多播通信中。由于网络编码是沿子图中的所有路径对信息进行转发，所以一个 k - 连通子图可以容忍多达 $k-1$ 条（个）链路（节点）故障——这对单播通信和多播通信来说都成立，而且不需要考虑计算多条链路/节点不相交路径的复杂度。

6.6.4 提供低延迟链路层编码

使用前向纠错（FEC）可以提高传输的可靠性。我们对数据包的编码应满足：对每组待传输的 K 个原始数据包都构造 N 个编码数据包并发送。通过构造码字我们可以实现：接收端只要接收到 K 个或更多的编码数据包就可以重建原始的 K 个数据包，而不需要关心接收到的具体是哪些数据包。我们基于丢包率、可靠性要求和延迟约束来选取 K 和 N 的值。如果在源节点和目的节点之间采用端到端编码（比如 NORM[1] 或数字喷泉码[15]），那么其传输容量就等于该路径的每段链路上的数据包成功传输概率的简单乘积。考虑图6-9所示的两跳路径，其传输容量为 $(1-\varepsilon_1)(1-\varepsilon_2)$，其中 ε_i 表示第 i 段链路上的丢包率（为简单起见，假设所有链路容量均相同）。如果我们逐段链路进行 FEC 编码并在每个中继节点上都重建数据包，那么就可以把传输容量提高为 $\min(1-\varepsilon_1, 1-\varepsilon_2)$，但这会导致延迟增大，因为我们在转发之前需要对数据包进行重建。

网络编码可以实现最优的割集容量（对于图6-9，该容量为 $\min(1-\varepsilon_1, 1-\varepsilon_2)$），

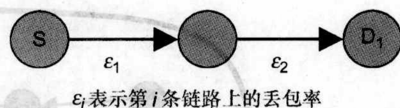


图6-9　网络编码技术将低延迟链路层编码作为自身操作的一部分

而且在中间节点上无需重建数据包，也就是说，网络编码具有端到端编码的低延迟性，因为它只需要在目的节点上对数据包解码。

MANET 的链路丢损特性变化过快导致我们不能准确地描述链路。对于这种情况，我们需要在链路层采用一种无速率编码方案，即该方案可以持续产生编码数据包，直至目的节点能够恢复出原始数据包为止。在多播网络中，诸如数字喷泉码等的编码方法在端到端之间实现了上述无速率方案。而网络编码则在跳与跳之间提供了这个功能——每个节点都持续发送随机线性组合直至下游节点接收到足够的信息——其预期效益是降低传输次数并提高容量。我们需要通过反馈来让上游节点知道什么时候停止发送以及以什么速率发送。6.5 节中详细讨论了实现上述功能的算法。

6.6.5　极端机会路由

Biswas 和 Morris[16] 观察到：我们一般难以提前知道哪个节点会接收到某次无线传输。因此为数据包预先指定目的节点的传统路由方案是低效的——更好的做法是先发送数据包，在知道哪些节点接收到了这个数据包之后再决定由哪些节点来转发它。基于上述观察结果，Biswas 和 Morris 提出了极端机会路由协议（ExOR）。该协议基于实际接收到了数据包的节点的信息动态地确定转发每个数据包的节点序列。该协议选择距离目的节点最近的节点（使用预期传输次数（ETX）度量距离）来传输数据包。相比于通过最优可能预定路径传输，该协议能够将数据包运送得更远（平均来说）。该算法侧重于面向单播通信，在对多播通信的支持方面不易扩展。

参考文献［16］中给出了两个启发性的例子来说明 ExOR 带来的增益。接下来将讨论如何把机会路由纳入网络编码——进而扩展到多播通信中。

6.6.5.1　长跳

节点接收到某次传输的概率通常是发送机与该节点之间的距离的函数。对于这种情况来说，距离发送节点近的邻居节点接收到传输的概率较高，而距离远的邻居节点接收到的概率较低。典型的路由协议要求我们只能使用那些质量足够好的链路（因为链路质量差会导致链路层的重传次数过多）。这意味着我们不会使用那些偶尔可以把数据包移动较长距离的链路——从而导致网络的传输效率降低。由于 ExOR 协议是在节点接收到数据包之后才确定传输顺序，因此它能够利用这些低概率长链路的优势（见图6-10）。

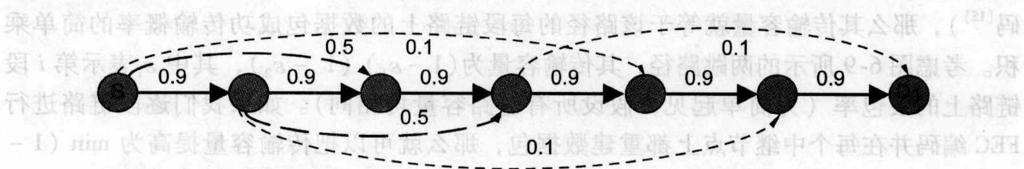


图6-10　网络编码与机会接收可以自然结合，因为节点只要接收到新生的数据包就会将它存储到本地

网络编码能够以一种自然的方式支持对长跳的利用。如果图6-10中的所有节点都属于某个子图，那么可以利用子图节点之间的链路质量来确定每个节点参与转发的速率。当一个较远的节点机会接收到一个网络编码数据包时，它可以简单地把这个数据包添加到它存储的关于该分代的信息池中——这可以降低目的节点在解码之前所需等待的时间。需要注意的是，网络编码在单播通信和多播通信中都可以提供ExOR的“长跳”优势。

6.6.5.2　大量的有损链路

传统的路由协议会从所有可能通往目的节点的下一跳节点中选择一个“下一跳”。通常来说，这些节点中与目的节点之间“距离”最短的节点会被选定为下一跳节点。如果可选的下一跳节点数量很多——但所有节点的质量都很差——那么上述策略就不那么理想了。例如在图6-11所示的场景中，源节点S和目的节点D_1之间有5个可选的下一跳节点。假设每个节点成功接收的概率都是20%，那么虽然任何一个节点都不太可能成功接收到某次传输，但是这5个下一跳节点中至少有一个可以成功接收的概率是非常大的。

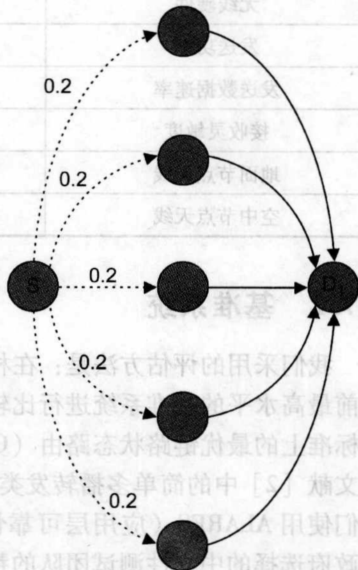


图6-11　网络编码的子图构造需要考虑链路质量，如果合适的话，它可能会使用多条低质量链路

ExOR在节点接收到数据包之后才确定由哪个节点继续发送，通过这种方式它可以充分利用多条有损链路。对于图6-11中所示的场景，ExOR将利用所有的中间节点来进行转发。网络编码也可以用同样的方式来利用多条有损链路——不同的是，网络编码在多播通信中也可以做到这一点。

总而言之，迄今出现的用于解决单播、多径（多播）路由、链路层编码和机会转发这些不同的问题的许多不同的算法都可以被纳入网络编码中。

6.7　现场实验的基础设施

我们的实验目的是在一个被设计为支持采用便携式嵌入式平台的移动节点的MANET环境中，通过一系列的现场实验来评估在802.11上运行的CONCERTO协议集的实际性能（并与一个基准协议集进行比较）。

本节将描述现场实验中使用到的硬件、软件应用程序和工具，以及用于评估现场实验性能的相应标准。此外，还简要介绍了用于对比的基准协议集。

6.7.1　硬件

在整个实验过程中我们尝试了3个平台，分别是：Avilla平台——采用533MHz的

单核 ARM 处理器；Aeon 平台——采用 2GHz 的双核英特尔处理器；以及 Pismo 平台——采用 1GHz 的双核 ARM 处理器。在所有被记录的实验中我们使用的都是 Pismo 平台，因为它能够提供 CPU 能力和低功率消耗（8W）之间的良好权衡，而不需要特殊的冷却处理。此外，我们还采用了商用现成的（COTS）基于 Atheros 芯片组的 802.11a/b/g 无线射频标准，其特性参数见表 6-1。

表 6-1　802.11 无线射频标准的特性参数

无线硬件	通用的 802.11 无线网卡（XR-2）
发送功率	600mW（28dBm）
发送数据速率	11Mbit/s
接收灵敏度	-92dBm
地面节点天线	12″全向（+7dBi）
空中节点天线	12″全向（+7dBi）加 120°扇区（+9.5dBi）

6.7.2　基准系统

我们采用的评估方法是：在相似的网络和通信条件下将提出的 CONCERTO 系统与目前最高水平的基准系统进行比较。该基准系统（见图 6-12）由运行在 802.11 无线射频标准上的最优链路状态路由（OLSR）协议[4]、基本多播转发（BMF）协议[17]（与参考文献［2］中的简单多播转发类似）以及基于否认的可靠组播（NORM）协议[1]构成。我们使用 ALARES（应用层可靠性系统[18]）来恢复在网络长期中断期间丢失的数据。在政府选择的中立性测试团队的帮助下，我们对这些协议进行了选择和调整。

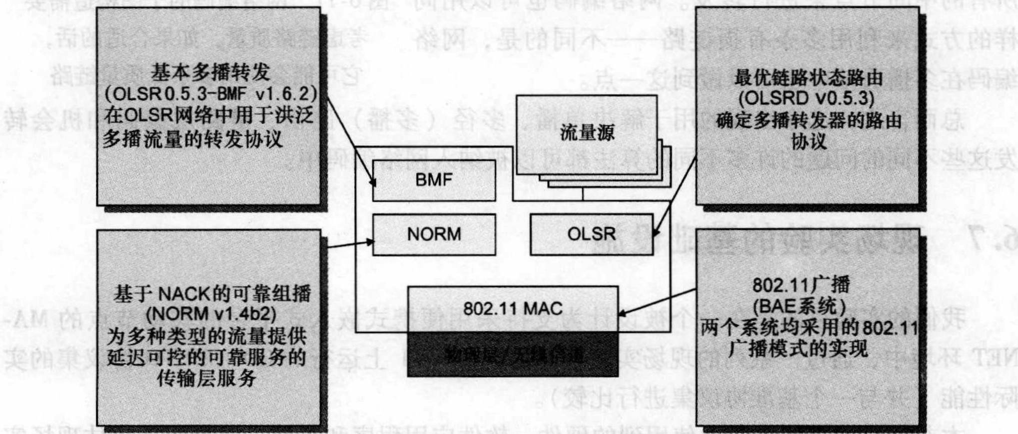


图 6-12　基准系统架构

OLSR 是一个针对 MANET 设计的高效的 IP 路由协议。采用该协议时，每个节点都产生并洪泛拓扑信息，该信息中列出了该节点的所有邻居节点以及与每个邻居节点之间

的链路质量。与其他路由协议不同的是，OLSR 仅使用所有节点的一个子集进行洪泛（称为多点中继），这可以提高网络容量的利用效率。

NORM 协议的设计目的是为一般的 IP 多播路由及转发服务提供对大量数据对象或数据流的端到端的可靠传输。NORM 采用选择性否认（NACK）机制来保证传输的可靠性。该协议使用了基于 FEC 的修复策略，它允许通过一次重传帮助多个目的节点修复而不需要考虑这些节点具体丢失了哪些数据包。我们可以选择性地采用 FEC 来提供传输的前摄鲁棒性。视频应用的延迟要求（4s）与 NACK 帧的使用不兼容，因此对于视频应用来说，其 NORM 协议仅配置了符码率为 1/2 的 FEC 编码。而对于文件传输来说，其 NORM 协议配置了符码率为 1/2 的 FEC 编码并使用了 NACK 帧。

ALARMS 的设计目的是以一种与 FEC 和 NORM 互补的方式为 MANET 提供更长期的可靠性。ALARMS 主要针对的是信源节点对于目的节点不可达的临时网络分区（例如 NORM 无法工作的情况）。ALARMS 在每个节点上都会存储该节点监听到的传输，并在源节点不可达时选择一些修复服务节点来给目的节点提供信息。

对于不同类型的信源，其可靠性机制配置如下：

CONCERTO 系统

视频	聊天	SA	FX
半可靠 CONCERTO 转发	半可靠 CONCERTO 转发	每秒发送一次更新信息	全可靠 CONCERTO 转发

基准系统

视频	聊天	SA	FX
符码率为 1/2 的 NORM 协议，无 NACK 帧	每个数据包发送 10 次，采用 ALARMS 协议	每秒发送一次更新信息	符码率为 1/2 的 NORM 协议，采用 NACK 帧，采用 ALARMS 协议

6.7.3 通信方案

我们在相似的条件下同时使用 CONCERTO 和基准系统来进行所有的现场实验。我们使用一个脚本工具（“Nettion”[19]）来控制现场实验，以确保 CONCERTO 和基准系统采用的通信模式相同。CenGen 公司为视频、聊天、态势感知（SA）和文件传输（FX）提供了相应的软件应用程序，这些程序产生或重播实时应用程序通信流或录制的视频流。我们在应用程序通信流中夹带元数据以测量各种指标，比如丢包率和后处理时延。不同信源的数据速率如下：

应用程序通信流

视频	聊天	SA	FX
100kbit/s MPEG4，12 包/s，1024B/包	2 ~ 400B/包，平均 1 包/30s	200B/包，平均 1 包/s	文件的平均大小为 1MB

6.7.4 评估方法

我们的目标是在实际环境中根据终端用户的效用来测试和评估 CONCERTO 系统，并将它的性能与基准系统进行比较。效用的定义基于不同的应用程序，并需考虑延迟和丢包率的影响。视频的效用指标需把握的一个概念是：假设发送一段 10s 的视频块，如果视频接收者能够在小于4s 的端到端延迟内至少接收到该块的90% 的数据包，那么我们就认为接收者对这段视频是满意的。我们还观察到，虽然对于移动多跳网络中的较长的距离来说，视频流的传输较困难，但其效用却较高。为了更好地阐述 MANET 中的“传输效用”这个概念，我们设计了一个距离 - 效用指标，该指标定义为每对源节点 - 目的节点之间的距离与其视频效用的乘积的和。

文件传输的效用指标需把握的一个概念是：如果文件传输的接收者能够在确定的延迟期限内接收到完整的文件，那么我们就认为该接收者是满意的。文件传输的效用指标指的是在不同的文件传输中感到满意的接收者所占的平均比例。由于文件大小的变化并不显著，因此我们可以把所有文件的传输延迟都限定在 15min 内。

每条聊天信息都会被发送给所有的目的节点。如果一个数据包在发出之后的 2s 内被接收，那么我们就认为接收该数据包的“效用值”是 100%；否则该效用值将随传输时间的增加指数递减，其衰减时间常数为 3h。聊天场景的效用值等于接收到的所有数据包的效用值之和除以接收次数。

每个态势感知（SA）信息都会被发送给所有的目的节点。如果一个 SA 应用程序能够每 5s 收到一个更新信息，那么我们就认为该应用程序是满意的。如果应用程序没收到更新信息的时间超过 5s，则其效用值会随时间的增加指数递减，衰减时间常数为 100s。SA 场景的效用值等于该阶段考虑的所有时间片段的效用的平均值。

6.8 实验结果及分析

本节将给出在动态移动战术场景下 CONCERTO 系统相比于基准协议的性能效益的结果和分析。此外，还将给出在这些战术场景的若干工程变体下的实验结果，这些结果可以帮助读者理解我们的系统在更具一般性的环境下的潜力。

6.8.1 实验场景

为了在无线移动 Ad Hoc 网络中展示 CONCERTO 系统的性能，我们设计了一个包含 35 个无线节点的战术场景（见图 6-13）。其中 31 个节点采用电池供电，每个节点都放置在便携背包中由单人背负，并被分为 3 队（Alpha、Bravo 和 Charlie）；另外 4 个无线节点中的 2 个安装在两辆卡车上，2 个安装在两架飞机上。这两辆卡车用于部署并集合全体人员往返于中央指挥所。

在该场景中模拟的事件是有 3 个地点分别受到来自 3 个分队的攻击。在模拟方案的初始阶段，军队在指挥所附近的着陆区（LZ）聚集，并派出一个小队巡逻指挥所。分



图6-13 在Alpha/Bravo阶段，场景中有10个单兵携带的无线电台、2个安装在卡车上的无线电台、2个安装在飞机上的无线电台（图中没有画出）以及21个静止无线电台

队成员首先在关键位置部署静态节点，接着包围他们的目标站点。接下来各分队开始对他们的目标站点发起"进攻"，进攻方式是围绕目标站点走动。进攻过程可以分为两个阶段：在Alpha/Bravo阶段，Alpha分队和Bravo分队围绕各自的目标走动，而Charlie分队的无线设备静止不动；而在Bravo/Charlie阶段，Alpha分队的无线电台静止不动。车辆在道路上来回行驶，运送人员和物资。表6-2中列出了该战术方案的不同阶段；图6-13中所示的是该方案的Alpha/Bravo阶段。

表6-2 CONCERTO系统和基准系统的性能评估方案的各个阶段

阶段	行为	时间/min
LZ	在着陆区（LZ）聚集	0~30
部署	部署Alpha、Bravo和Charlie 3个分队	30~40
Alpha/Bravo	Alpha分队和Bravo分队发起进攻	40~70
	调整人员	70~110
Bravo/Charlie	Bravo分队和Charlie分队发起进攻	110~140
工程	压力测试	140~150

我们将执行两种方案：地面方案不使用飞机，而空中方案使用了两架在空中盘旋的飞机。每架飞机都配备了1个无线电台和2根高增益天线。这两架飞机作为中继能够为地面部队之间提供更短的通信路径。

上述场景中存在大量的低质量链路。图6-14所示的是地面方案在Bravo/Charlie阶段的拓扑结构。图中的链路的两种颜色分别表示在该链路的两个端点上测量出的链路质量。需要注意的是，图中存在大量的丢包率大于30%的链路和非对称链路。

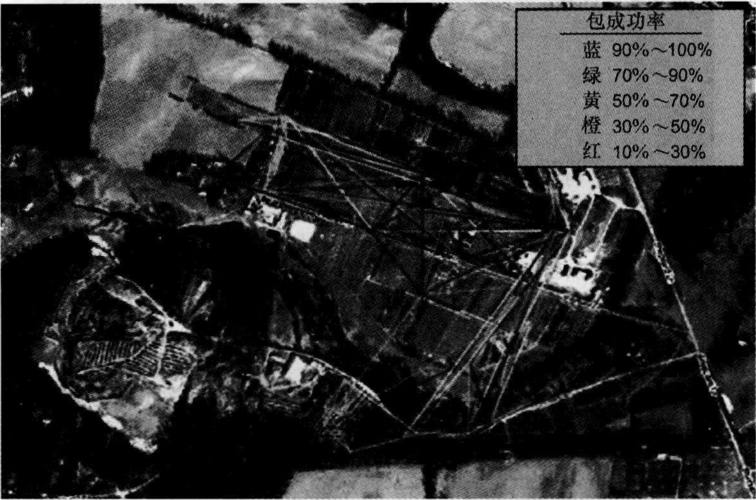


图 6-14　地面方案在 Bravo/Charlie 阶段的拓扑结构。注意图中有大量的低质量链路

我们通过让不同的源节点将 100kbit/s 的 MPEG4 格式的视频流多播给不同的包含 4 个目的节点的节点组来产生视频流量。为了测试 CONCERTO 系统和基准系统的最大可承受负载，在 LZ 阶段、Alpha/Bravo 阶段和 Bravo/Charlie 阶段应用了 3 种视频负载，见表 6-3。视频负载 1 中的 3 个视频会话的源节点分别是各队的 1 个队员，目的节点组则是同一队的其余 4 个队员。由于同一分队的各成员之间的通信通常在一跳以内，因此视频负载 1 是一个相对较轻松的方案。在视频负载 2 和视频负载 3 中，额外添加的源节点把信息发送给其他分队的 2 个成员以及 2 个位于其他较远位置的成员——这是一个更有压力的多跳方案。表 6-3 中的第 3 列表示所有目的节点能够接收到的视频流量的总和。由于每个视频会话都有 4 个目的节点，因此总接收流量应是视频会话个数与视频带宽 (100kbit/s) 乘积的 4 倍，这也是 MANET 在理想工作的条件下能够接收到的总流量。

表 6-3　用于评估性能的视频负载

视频负载	测试内容	总接收流量/ (kbit/s)
负载 1	3 个队内视频会话	1200
负载 2	视频负载 1 外加 1 个队间视频会话	1600
负载 3	视频负载 2 外加第 2 个队间视频会话	2000
负载 4（压力测试）	视频负载 3 外加第 3 个多跳视频会话	2400
负载 5（压力测试）	视频负载 2 外加第 4 个多跳视频会话	2800

在工程压力测试中我们将添加额外的视频流，这样设计的目的是为了确定 CONCERTO 协议的工作上限。每个额外视频流都有 4 个目的节点并需经历多跳传输。

图 6-15 中示意了各个视频流。在 3 个目标站点附近的 3 个簇内被圈出的节点是视频负载 1 中的 3 个源节点。橙色圈出的是视频负载 2 中的额外添加的源节点，橙色的线

指示了该额外视频会话的各个目的节点。同样地，紫色圈出的节点和紫色的线表示视频负载3中的额外负载。需要注意的是，视频负载2和3的额外会话包含了一些比视频负载1要远的目的节点。

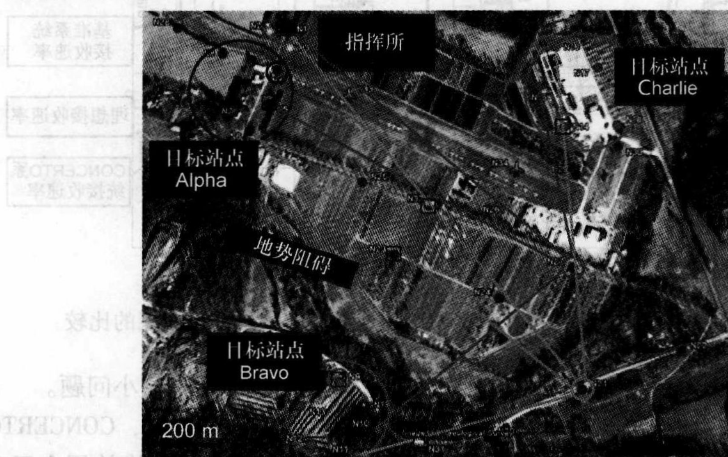


图6-15 在地面方案和空中方案中用于评估基准系统的CONCERTO系统性能的视频流。视频负载1位于“小队内”，而视频负载2和3添加了“小队间”的多跳会话（分别用橙色和紫色表示）

我们在7个节点之间进行文件传输（FX）（每簇有1个FX源节点、2个卡车节点、1个指挥所节点和1个感知节点）。每个FX节点轮流将文件传输给所有其他FX节点。所有节点都产生并接收聊天流量和SA流量。

6.8.2 实验结果

在所有的实验中，我们都进行了视频、聊天、SA和文件的混合传输，其中聊天和SA在所有流量中仅占一个相对较小的比例（<10%）。由于聊天和SA采用了鲁棒的应用级可靠性算法，因此在实验的各个阶段，在基准系统和CONCERTO系统下它们都表现出了较高的平均效用（大于90%）。下面讨论视频传输和文件传输的实验结果。

6.8.2.1 地面战术场景

图6-16中总结了基准协议（上方）和CONCERTO协议（下方）在地面方案中的视频性能。图中的蓝线表示性能理想时，可能接收到的视频总流量（见表6-3）。红线是所有目的节点实际接收到的视频总流量。前面提到，如果目的节点在小于4s的延迟内接收到“块”中的90%的数据包，那么其视频效用就定义为1.0，否则效用是0.0。在两个协议集下不同“块”的平均视频效用在图中用绿色表示。浅黄色方框中给出了不同视频负载的平均视频效用。

在实验的初始阶段（LZ阶段）所有节点都是静止的且呈网状结构。此时在3种视频负载下CONCERTO系统和基准系统的视频效用指标都接近理想（平均为99%）。在下一阶段，我们对小队成员和固定节点进行了部署，此时CONCERTO系统的表现（平均效用94%）要略好于基准系统（平均效用82%）。随着节点逐渐向部署的位置分散，

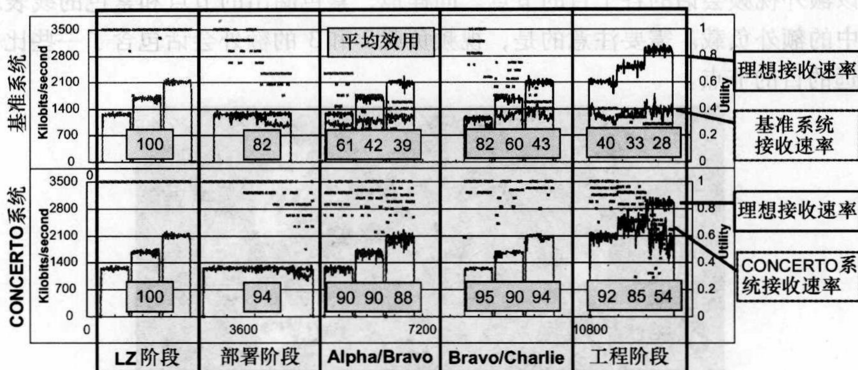


图 6-16　CONCERTO 系统和基准系统在地面方案中的视频性能的比较

即使对于视频负载很小（负载 1）的情况，基准系统也显现出了一些小问题。

在其余的阶段，CONCERTO 系统也表现良好：在所有的测试中，CONCERTO 的平均效用都介于 88% ~95% 之间，而基准系统就没有这么有效了（平均效用介于 30% ~60% 之间）。CONCERTO 系统性能良好的主要原因是：网络编码子图能够容忍网络节点的移动性；此外，CONCERTO 系统的端到端可靠传输机制能够有效地利用网络带宽，从而支持更多的视频会话。

在视频延迟方面，CONCERTO 系统和基准系统（见图 6-17）在所有方案下的测试结果都是不相上下的。在某些阶段中 CONCERTO 系统的延迟高于基准系统，这是因为我们测量的是目的节点接收到的数据包的延迟，而在实际测试中基准算法丢弃了大部分数据包。CONCERTO 系统的较高的延迟与其优于基准系统的性能相关。

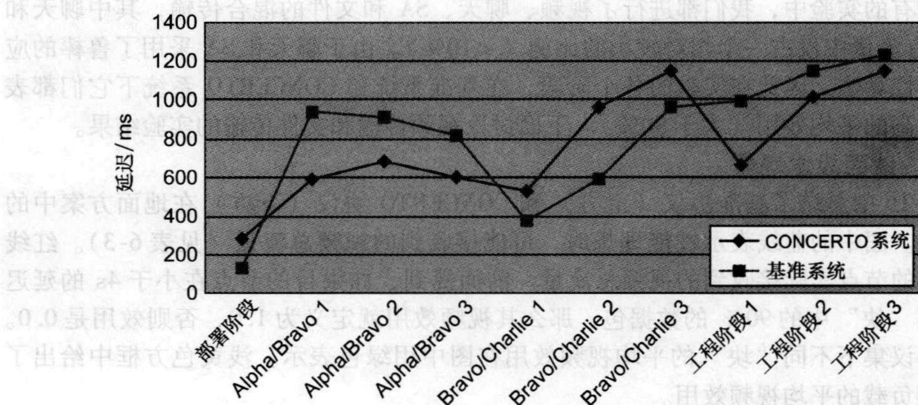


图 6-17　CONCERTO 系统和基准系统在空中方案下的延迟

此外，伴随这种高性能的是更低的网络负载。图 6-18 所示的是应用前面提到的 3 个级别的视频负载时，基准系统和 CONCERTO 系统在 Alpha/Bravo 阶段中表现出的性能。蓝

色实线表示源节点产生的视频总流量，绿色实线是 CONCERTO 系统的总传输量，而红色实线是基准系统的总传输量。由于视频流需要经历多跳传输（也就是说每个数据包都会被多次发送），因此总传输量会大于注入网络的视频流量。然而请注意，CONCERTO 系统产生的总流量要比基准系统少 2 ~3 倍。图中浅绿色的点和橙色的点分别表示 CONCERTO 系统和基准系统的平均视频效用。我们注意到虽然 CONCERTO 系统产生的流量较少，但它的性能反而更好：CONCERTO 系统的效用可达到 90%，而基准系统只有 47%。

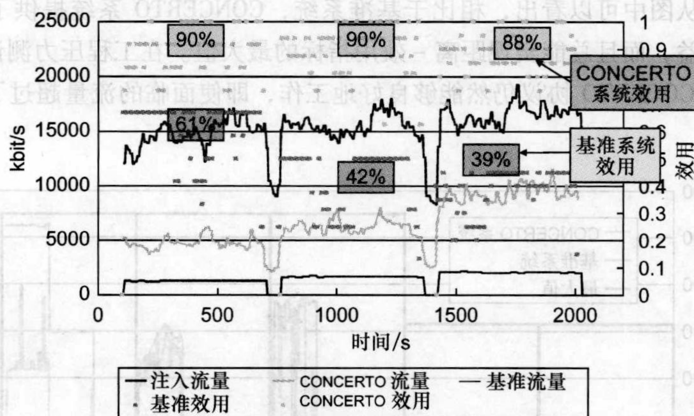


图 6-18 网络传输总量的比较

CONCERTO 系统的一个非常重要的好处是：不论距离远近都能够提供高视频效用。图 6-19 所示的是目的节点测量出的平均视频效用，它是目的节点与源节点之间的距离的函数。我们观察到 CONCERTO 系统和基准系统之间有一个明显的区别：不论距离多少前者都可以提供令接收者满意的视频，而后者的性能仅当目的节点位于 1 跳或 2 跳距离内时才是令人满意的。造成上述区别的主要原因是，CONCERTO 系统结合了网络编码和内置于网络编码转发算法中的逐跳可靠性。

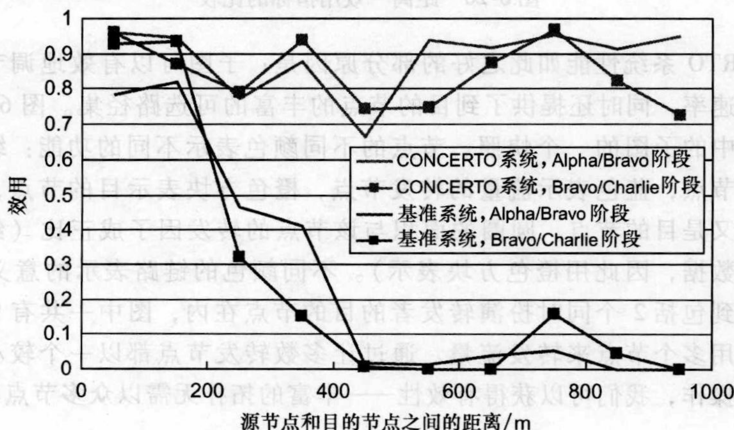


图 6-19 不同距离下的性能

我们引入距离-效用指标来量化不同距离下的视频传输效益，该指标定义为每个目的节点的效用值与源节点-目的节点之间的物理距离的乘积，再对所有目的节点求和所得的值。当所有的目的节点都以理想的效用接收流量时，其距离-效用指标取到最大值。图6-20所示的是（地面方案中）距离-效用指标的最大值以及基准协议和CONCERTO协议的实际测试值。在我们的战术实验中，视频负载1仅需经过“1跳”就能到达分队内的接收端，而视频负载2和3需要经过多跳，这使得它们的视频距离-效用值大大增加。从图中可以看出，相比于基准系统，CONCERTO系统提供了高达7倍的距离-效用增益，而且总能实现距离-效用指标的最大值。在工程压力测试中基准协议发生崩溃，而COCERTO协议仍然能够良好地工作，即使面临的流量超过了能够处理的范围。

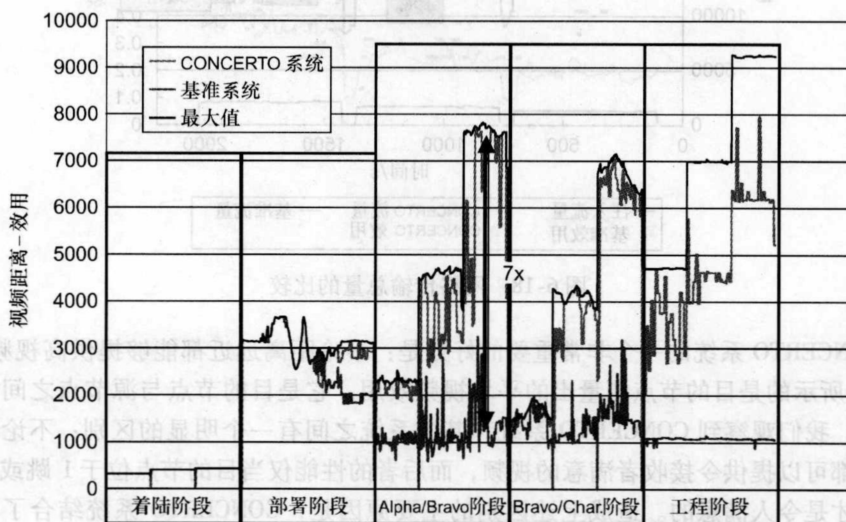


图6-20 距离-效用指标的比较

CONCERTO系统性能如此之好的部分原因是：子图可以有效地调节各个节点转发流量的速率，同时还提供了到目的节点的丰富的可选路径集。图6-21所示的是地面方案中的子图的一个快照，节点的不同颜色表示不同的功能：绿色的是多播会话的源节点，蓝色表示流量的转发节点，橙色方块表示目的节点，黄色的既是转发节点又是目的节点。圆圈的面积与该节点的转发因子成正比（纯目的节点不转发任何数据，因此用橙色方块表示）。不同颜色的链路表示的意义与图6-14相同。注意到包括2个同时扮演转发者的目的节点在内，图中一共有9个转发节点，子图使用多个节点来转发流量。通过让多数转发节点都以一个较小的转发因子进行转发操作，我们可以获得有效性——丰富的拓扑无需以众多节点的高速率发送为代价。

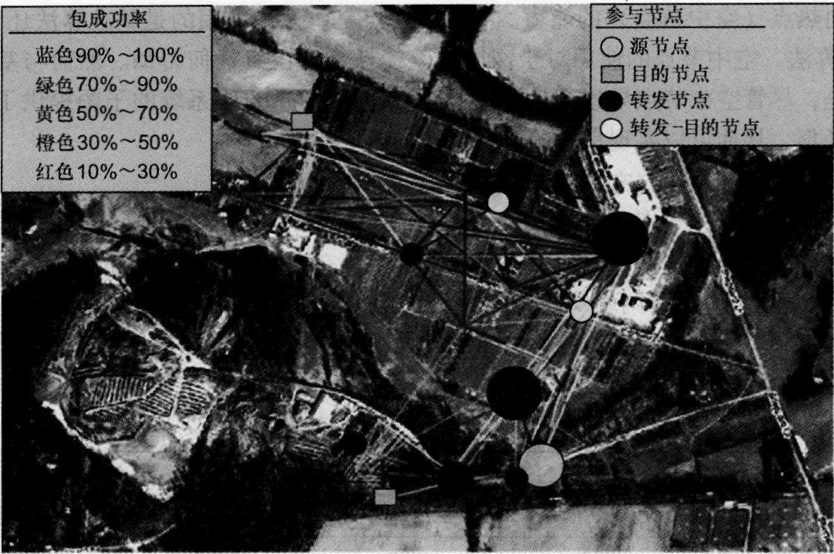


图6-21 由绿色（源）节点、蓝色（转发）节点、黄色（转发-目的）节点和橙色（目的）节点组成的子图。节点的大小表示其转发速率

6.8.2.2 空中战术场景的结果

有空中节点协助的移动战术场景的实验结果（见图6-22）与地面战术场景的结果相似。飞机的使用有助于减少多数源-目的节点路径的总跳数。但由于隐藏终端的碰撞问题（大部分地面节点都看不见对方但都能看见空中节点），从地面传输到空中节点的丢包率很高（50%~70%）。总体而言，在视频流量传输中，相比于基准系统，CONCERTO系统带来了显著的性能增益。基准协议仅能够在初始阶段和部署阶段提供良好的视频传输，而在空中战术场景的所有阶段，即使有大量低质量链路出现（如从地面到空中的大多链路），CONCERTO的表现也良好，其主要原因是网络编码子图具有将信

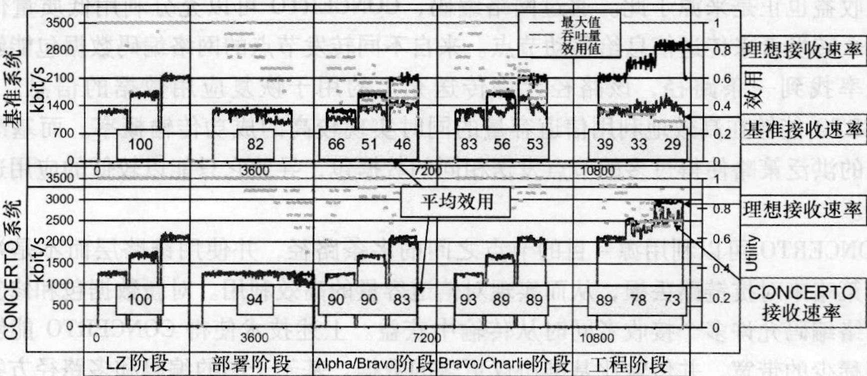


图6-22 空中方案的视频性能

息流引向热点（空中节点）的能力。有趣的一点是，无论采用的是基准算法还是 CONCERTO 算法，空中节点的使用对于网络的整体性能都没有任何帮助。然而与基准系统不同的是，尽管空中节点导致了热点的出现，但 CONCERTO 系统的子图提供了丰富的可选路径集，从而可以实现良好的性能。

6.8.2.3 战术场景下的文件传输结果

CONCERTO 在文件传输上获得了又一次重要的成功（见表 6-4）。CONCERTO 系统实现了接近 100% 的文件传输成功率，而基准系统在多跳战术场景下的文件传输成功率仅为 20% 左右。全可靠网络编码传输协议结合了对单个网络编码数据包分组（分代）的逐跳修复和对整个网络编码分组的端到端检测及修复策略，通过使用该协议可以实现对所有文件的成功传输。

表 6-4 不同阶段的平均文件传输效用 （文件传输结果）

	地面方案		空中方案	
阶段	CONCERTO	基准	CONCERTO	基准
着陆阶段	100%	100%	100%	100%
部署阶段	100%	47%	100%	47%
Alpha/Bravo	100%	26%	100%	16%
Bravo/Charlie	100%	21%	100%	20%

6.9 总结和展望

6.9.1 总结

在所有的阶段，尤其是视频需要经历多跳低质量链路的较为挑战的阶段，CONCERTO 系统均胜过了基准系统。CONCERTO 方案将网络编码和丰富的转发子图相结合，其性能收益也正是来源于此。通过网络编码，CONCERTO 可以充分利用低质量信道并使用多个转发者来传送信息给移动节点。来自不同转发节点的网络编码数据包能够以较高的概率找到一条路径，该路径允许传送充足的用于恢复应用数据的信息。因此，CONCERTO 能够在高效地利用信道容量的同时实现较高的成功传输概率。而基准协议集采用的洪泛策略使得过多的节点发送相同的数据包，导致它只能以较低的应用速率达到容量。

CONCERTO 可以利用源－目的节点之间的多条路径，并使用链路层而不是端到端的重传策略来对抗链路丢损，从而实现对信道容量的高效利用。对新数据包和修复数据包的网络编码允许多个接收者同时从传输中获益。上述技术使得 CONCERTO 能够有效地利用稀少的带宽，并传输比基准协议更多的数据。基于子图的编码和多路径方案可以提高 CONCERTO 系统在动态环境中对于个别链路和节点丢损的鲁棒性。

6.9.2　展望

虽然 CONCERTO 系统已经在数据包传输性能和带宽效率上获得了显著的增益，但是它在一些方向上仍有进一步改善的潜力。

子图构建算法的计算复杂度为 $O(n^2)$，其中 n 是节点数量。我们依赖于 OLSR 来发现并传播完整的网络拓扑信息，进而启动在每个多播会话的源节点上的路由子图计算。另一个更具扩展性的方案是仅利用本地拓扑信息并基于与距离矢量型路由算法类似的增量路径传播。在一个相关的项目中（由 ONR 资助的 BRAVO 项目），我们正在基于利用势场[20]的信息传播模型探索这样的算法。

尽管我们相信 CONCERTO 的性能优势独立于 MAC 层和物理层，然而如果把网络编码和新的 MAC 层协议以及允许接收多个数据包的基于多用户检测[21]的专用物理层硬件相结合，可能还会额外获得显著的性能增益。我们打算设计一个这样的解决方案，并验证将多数据包发送（通过网络编码）和多数据包接收结合带来的好处。

6.10　致谢

作者十分感谢 BAE 系统团队对 CONCERTO 项目作出的贡献（2006 年 8 月—2009 年 8 月）：Jennifer Costello、Brendan Coyle、Brian DeCleene、Victor Firoiu、Robert Flynn、Sean Griffin、Mark Keaton、Clifton Lin、Greg Lauer、Ryan Metzger、Soumendra Nanda、Tom Porcher、Sean Shen、Joseph Sivak、Charles Tao。

此外，我们还要感谢以下机构作出的贡献：加州理工学院、美国康乃尔大学、麻省理工学院、宾夕法尼亚州立大学、Stow Research、伊利诺伊大学香槟分校以及麻省大学阿模斯特分校。

参考文献

[1] B. Adamson, C. Bormann, M. Handley, and J. Macker, Negative - Acknowledgement (NACK) - Oriented Reliable Multicast (NORM) TransportProtocol, RFC 5790, November 2009.

[2] Simplified Multicast Forwarding, SMF, IETF draft, draft - ietf - manetsmf - 11.

[3] Open Shortest Path First, OSPF, IETF RFC 2328.

[4] T. Clausen, and P. Jaquet, Optimized Link State Routing Protocol (OLSR), IETF, RFC 3626.

[5] Protocol Independent Multicast, IETF RFC 4601.

[6] R. Ahlswede, N. Cai, S. Li, and R. Yeung, Network Information Flow, IEEE Transactions on Information Theory, 46: 1204 - 1216, 2000.

[7] S. Li, R. W. Yeung, and N. Cai, Linear network coding. IEEE Transactions on Information Theory, 49: 371 - 381, 2003.

[8] R. Koetter and M. Medard, An Algebraic Approach to Network Coding, IEEE/ACM Transactions on Networking, October 2003.

[9] T. Ho, R. Koetter, M. Medard, D. R. Karger, and M. Effros, The Benefits of Coding over Routing in a Randomized Setting, In Proceedings of 2003 IEEE International Symposium on Information Theory, June 2003.

[10] D. Lun, M. Medard, R. Koetter, M. Effros, On coding for reliable communication over packet networks. In Proc. 42nd Annual Allerton Conference on Communication, Control, and Computing, Sept. - Oct. 2004, invited paper.

[11] D. S. Lun, Ratnakar, N., Medard, M., Koetter, R., Karger, D. R., Ho, T., Ahmed, E., and Zhao, F., Minimum - cost multicast over coded packet networks, IEEE Transactions on Information Theory, Vol. 52, Issue 6, pp. 2608 - 2623, June 2006.

[12] S. Chachulski, M. Jennings, S. Katti, and D. Katabi, Trading Structure for Randomness in Wireless Opportunistic Routing, SIGCOMM' 07, Kyoto, Japan, pp. 169 - 180.

[13] B. Cain et al., Internet Group Management Protocol, Version 3, IETF RFC 3376, October 2002.

[14] J. Park, D. Lun, Y. Yi, M. Gherla, and M. Medard, CodeCast: A network coding based ad hoc multicast protocol, IEEE Wireless Comm. Mag., 2006.

[15] J. Byers, M. Luby, M. Mitzenmacher, and A. Rege, A Digital Fountain Approach to Reliable Distribution of Bulk Data, Proc of ACM SIGCOMM' 98.

[16] S. Biswas and R. Morris, ExOR: Opportunistic Multi - hop Routing for Wireless Networks, SIGCOMM' 05, Philadelphia, USA, pp. 133 - 144.

[17] Basic Multicast Forwarding, BMF, http: //sourceforge. net/projects/olsrbmf/.

[18] Application LAyer REliability System, Alares, http: //www. darkcornersoftware. com/confluence/. display/ALA/Alares.

[19] Nettion, NETwork TestIng and Operational eNvironment, http: //www. darkcornersoftware. com/nettion. html.

[20] V. Firoiu, and H. Liu, BRAVO: Potential Field - Based Routing and Network Coding for Efficient Wireless MANETs, in preparation.

[21] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard, and J. Crowcroft, XORs in the Air: Practical Wireless Network Coding, SIGCOMM' 06, Pisa, Italy, pp. 243 - 254.

第7章　安全网络编码：保密与可靠通信中的界和算法

Sidharth Jaggi 香港，香港中文大学，信息工程系

Michael Langberg 以色列，赖阿南纳，以色列开放大学，数学与计算机科学系

摘要

网络编码允许网络路由器在转发它们接收到的数据包之前，对其中的信息进行混合。理论证明，这样的混合对提高吞吐量和鲁棒性是非常有益的。然而，当网络中存在恶意节点时，情况会怎样呢？为了窃听正在进行的通信，这些恶意节点会“连接”到网络中，假装转发的是来自源节点的数据包，但实际上它们会将损坏的数据包注入信息流中从而破坏通信。

由于网络编码允许路由器混合数据包的内容，因此一个损坏的数据包能够破坏去往目的节点的所有信息。除非解决这个问题，否则当网络中存在上述恶意对手时，网络编码转发的性能可能反而会比直接转发还差。

本章将研究存在被动窃听节点和主动干扰节点的网络中，使用网络编码的多播通信。令人惊讶的是，研究结果表明，在所讨论的环境中采用一些分布式且计算有效的方案可以实现高速率的可靠保密通信。本章总结了在可靠保密通信网络这个活跃又有趣的领域近十年的研究成果，所给出的算法涉及对组合数学、线性代数、密码学和编码理论等工具的使用，并给出了若干范例。

关键词

网络编码、网络通信、纠错、窃听信道、分布式协议、密码学、容量。

7.1　引言

网络编码允许网络路由器在转发它们接收到的数据包之前，对其中的信息进行混合。理论研究证明，这样的混合能够最大限度地提高网络吞吐量[1,29,40,44]。在多播通信中可以通过一种低复杂度的分布式方法来完成，且对网络中的丢包和故障其具有鲁棒性[24,46]。此外，最近在有线和无线环境中实现的网络编码也证明了它的实际效益[20,32]。

然而，当网络中存在恶意节点时，情况会怎样？这些恶意节点会“连接”到网络中，以窃听正在进行的通信。此外，一些节点还会假装转发来自源节点的数据包，但实际上将损坏的数据包注入信息流中。由于网络编码允许路由器混合数据包的内容，因此一个损坏的数据包能够破坏去往目的节点的所有信息。如果不解决这个问题，那么当网络中存在上述恶意对手时，网络编码转发的性能可能反而会比直接转发还差。

本章将研究存在被动窃听节点和主动干扰节点的网络中，使用网络编码的多播通信。令人惊讶的是，研究结果表明，在所讨论的环境中采用一些分布式且计算有效的方案可以实现高速率的可靠保密通信。如果不考虑分布式网络编码引入的复杂度，可以看

到，关于在点到点链路中进行可靠保密通信的许多经典研究结果有着直接类似的网络设置。本章总结了在可靠保密通信网络这个活跃又有趣的领域近十年的研究成果。

本章内容由七小节组成[⊖]。7.2 节对全章使用到的符号、定义和模型进行了设定。7.3 节中研究了存在被动对手的网络中通信的情况，这里的被动对手只有窃听能力且试图获取网络中传输的信息。对于这种情况，我们的目标是设计保密通信方案，即可以阻止对手了解源节点所发送信息的方案。7.4 节中考虑了网络中存在主动对手的情况，这里的主动对手同时具有窃听和干扰能力，其目的是让终端节点错误解码。在这种情况下成功的通信就意味着可靠通信，也就是正确解码。在 7.5 节中，我们再次考虑网络中存在能够同时窃听和干扰的主动对手的情况，但这次的目标是设计保密且可靠的通信方案。最后，在 7.6 节中简单地关注了其他一些不能归入上述分类中的模型，并在 7.7 节中对本章内容进行了总结。

在以上每节中的概述都包含了两到三个细化模型。首先考虑的是相干场景，假设其中的终端节点知道网络的拓扑信息以及所使用的通信方案；其次考虑的是非相干场景，该场景中假设终端节点没有关于网络拓扑和所使用码字的信息。对于上述两个场景均遵循信息论分析，假设对手的计算能力是无限的，并且了解关于网络拓扑以及正在使用的通信方案的完整信息。需要特别强调的是，任何在非相干场景下可实现的速率 R 对于相干场景也是可实现的，且任何在相干场景下基于 R 的上界对于非相干场景也成立；最后还考虑了对手的计算能力有限的情况，并讨论了在一定的加密假设条件下的方案。在 7.4 节中进行了扩展讨论，7.3 节和 7.5 节中的加密方案可以简化为 7.4 节中讨论的内容。7.2 节中给出了对每个细化模型的详细说明。

7.2 模型

我们使用的是一个涵括有线网络和无线网络在内的通用模型。为了简化标识，只考虑从一个源节点到一个目的节点的通信问题。但与大多数网络编码算法相似的是，此技术可以推广到多播通信中。

7.2.1 威胁模型

源节点 Alice 通过一个有线或无线网络与接收节点 Bob 通信。在网络中的某处还藏有一个攻击节点 Calvin。Calvin 的目标是尽量阻止或减少从 Alice 传输给 Bob 的信息，并/或窃听该信息。它能够观察到部分或所有的传输并注入它自己的信息。当它要注入自己的数据时，会将这些数据伪造成从 Alice 到 Bob 的信息流的一部分。

Calvin 十分强大，对相干和非相干场景，均假设 Calvin 知道 Alice 和 Bob 的编码和解码方案，以及内部节点采用的网络码字。此外，它还知道确切的网络实现。除非特别指出，否则假设 Calvin 的计算能力是无限的。对于后一种情况，将精确地指定限制 Calvin 的计算问题（如离散对数问题）。

⊖ 译者注：原文是四小节，应是笔误。

7.2.2 网络和码字模型

1. 网络模型

除特别指出外，我们将网络建模为一个无环图。每次传输都沿从发送节点到观察节点的一条边运送一个数据包。图模型用于表征有线网络。对于无线网络，可以将网络建模为一个超图，图中的每条边都由瞬时信道实现来确定（由于衰落或碰撞，数据包可能会丢失），并将发送节点与所有监听到该传输的接收节点相连接。我们将聚焦于有线场景，但是有一些结果也可以自然地扩展到无线场景中。对于非相干场景，假设 Alice 和 Bob 在传输之前不知道其网络图，但在相干场景下则假设相反。

2. 源

Alice 产生不可压缩的数据，并希望通过该网络将这些数据传送给 Bob。要做到这一点，Alice 会依据使用的编码算法（在后续章节中描述）对其数据编码。

3. 对手

Calvin 可以控制网络中的某些链路，假设它能够破坏网络中 z_0 条链路上的任意子集传输的信息，并能够监听 z_1 条链路上的任意子集传输的信息，且 Alice 和 Bob 均不知道 Calvin 控制的链路集由哪些链路组成。此外，还假设在整个数据块通信的过程中，上述链路集不会发生任何更改。在 7.3 节中，假设 z_1 无限大，但 $z_0=0$（此时的目标是设计保密的通信方案）。在 7.4 节和 7.5 节中假设 z_0 取正值，并讨论 z_1 取不同值时的情况（此时的目标分别是设计可靠的通信方案和可靠且保密的通信方案）。

在上述模型中，假设对手 Calvin 施加的错误信息是被添加到网络中传输的源信息上。此外，还有一种模型是用错误信息覆盖 Alice 发送的已有信息。要强调的是，在 Calvin 掌握了链路中传输信息的条件下，这两种模型是等价的。用信息 z 覆盖信息 x 等价于将 $-x+z$ 添加到实现编码的域上。然而当 Calvin 不了解链路上传输的信息时，这两个模型可能会有所不同。

4. 接收节点

接收节点 Bob 通过后文中将讨论的解码流程对它接收到的信息进行解码。

5. 网络变换

在许多提出的方案中，网络中执行的都是经典分布式网络编码[24]。具体而言，内部节点发送的所有数据包都是它接收到的数据包的随机线性组合。因此，网络对目的节点的影响可以表示为 $\boldsymbol{Y}=\boldsymbol{TX}+\boldsymbol{T'Z}$，其中矩阵 $\boldsymbol{X}$ 表示被编码的源信息，矩阵 $\boldsymbol{Z}$ 表示网络指定的错误信息，矩阵 $\boldsymbol{Y}$ 表示终端节点接收到的信息，矩阵 $\boldsymbol{T}$ 和 $\boldsymbol{T'}$ 均表示由网络编码方案规定的线性变换。和大部分其他网络编码文献一样，假设编码是在给定的有限域 $\mathbb{F}$ 上进行。

定义 我们定义了如下概念：网络容量指的是在平均时间内能够从 Alice 传递给 Bob 的数据包的最大数量，用 C 表示。如果假设没有来自对手的干扰，则 C 等于网络的最大流，此外它还是从源节点到目的节点的最小割（对于多播情况，C 定义为源节点到所有目的节点的最小割的最小值）；错误概率指的是 Bob 不能准确地恢复出 Alice 的信

息的概率。速率 R 是表示单位时间步长内能够从 Alice 传递给 Bob 的信息符号的平均数量。如果对于任意的 $\epsilon_1 > 0$ 和 $\epsilon_2 > 0$，存在一个可对分组大小为 n 的数据包编码的编码方案，满足速率大于等于 $R-\epsilon_2$ 且错误概率小于等于 ϵ_1，则就说速率 R 是可实现的。给定攻击场景下的容量是指该场景中可实现的最大速率。

7.3 窃听安全

首先考虑在多播网络编码环境中的保密通信问题。

7.3.1 相干情况

存在一个隐藏的窃听节点的相干场景，该节点与网络中的 z_I 条链路相连接，考虑在该场景中进行保密通信（也称为安全通信）的速率。特别指出的是，我们用随机变量 X 表示源节点发送的信息，用 Y_i 表示终端 t_i 接收到的该信息，并用 Z 表示窃听对手监听到的该信息。与香农定义用于理想安全系统的条件[64]相类似，要求 $H(X \mid Y_i)=0$（这意味着终端节点可以推断出源节点信息）且 $I(X;Z)=0$（这意味着通信是安全的）。令 C 表示最大多播通信速率（存在一个窃听者的情况下），研究表明最优速率 $R=C-z_I$ 下的安全通信是可实现的。且此速率是最优的，因为即便是在单个源节点 s 希望通过 C 条 (s, t) 链路与单个终端节点 t 通信的单跳单播网络中，也不可能以更高的速率进行安全通信。以下标准论证可以证明这一结论。令随机变量 $\overline{Z}$ 表示链路中没有被窃听节点监测到的信息（因此有 $Y_i=(Z,\overline{Z})$）。那么有：

$$R=H(X)=H(X|Y_i)+I(X;Y_i)=H(X|Y_i)+I(X;Z,\overline{Z}) \tag{7-1}$$

$$=I(X;Z,\overline{Z})=I(X;Z)+I(X;\overline{Z}|Z) \tag{7-2}$$

$$=I(X;\overline{Z}|Z)\leqslant H(\overline{Z})\leqslant C-z_I \tag{7-3}$$

依据我们对安全通信设定的条件可知有等式（7-1）⇒（7-2）且（7-2）⇒(7-3)，依据信息论标准等式（不等式）以及 $\overline{Z}$ 最多包含 $C-z_I$ 条链路这一事实，可以推导出其余的等式（不等式）。

在过去的十年中，一些研究工作已经解决了相干网络编码环境下的安全通信问题。这些工作的主线起始于 Cai 和 Yeung 在参考文献［9］中的研究，他们考虑的是存在一个窃听者的网络中，通过加强使用线性网络编码方案来实现与保密节点之间的速率为 C 的通信。对一个已有的（如 Jaggi 等人在参考文献［29］中提出的）线性网络编码方案的加强是以一种端到端的方式来完成的。也就是说，网络的内部节点没有察觉到通信正在被监听，仍然使用原始的编码方案。可通过源节点的加强编码以及终端节点的专门解码来处理对手的窃听。

为了达到这个目的，源节点使用的加强编码方案包括两个步骤：首先源节点将 $C-z_I$ 个字符组成的信息 X 和由 z_I 个字符组成的均匀分布的随机向量 R 进行组合，得到 (X, R)；然后对 (X, R) 进行某种可逆线性变换 $\boldsymbol{T}$，得到长度为 C 的信息 $\boldsymbol{M}$，接着使用原始的（可能是非保密的）网络编码方案在网络中传输信息 $\boldsymbol{M}$。解码时，终端节点

首先恢复出 $\boldsymbol{M}$，然后通过 $\boldsymbol{T}$ 恢复出 X。

下面考虑参考试文献［9］中提出的保密方案。将原始编码方案看做是固定的，参考文献［9］中的码字设计涉及对矩阵 $\boldsymbol{T}$ 的设计，该矩阵将进一步定义发送消息 $\boldsymbol{M}=\boldsymbol{T}(X, R)$。特别指出的是，对矩阵 $\boldsymbol{T}$ 的设计需要满足：从任意 z_I 个使用原始网络编码方案对 $\boldsymbol{M}$ 进行线性编码所得到的线性组合中都不能获得 X 的取值信息，这是一个艰巨的任务。参考文献［9］中通过一种类似于纠错码的吉尔伯特（Gilbert）构造[18]的贪婪迭代方式来构造 $\boldsymbol{T}$。参考文献［9］中的分析使用的是一个大小为 $q \geqslant \binom{|E|}{z_I}$ 的域$\mathbb{F}$，q 是网络参数的指数（考虑到最多存在 $\binom{|E|}{z_I}$ 个可以被对手窃听到的包含 z_I 条链路的子集，且矩阵 $\boldsymbol{T}$ 必须能对抗所有可能的攻击情况，故得出这个关于 q 的下界）。参考文献［9］中还考虑了对于不完全保密的扩展。通过基本相同的构造和证明方法，可以证明当 $I(X;Z) \leqslant i$（而不是 $I(X;Z)=0$）时，速率 $C+i-z_I$ 是可实现的（这里假设 $i \leqslant z_I$）。

定理 7-1[9]　令 G 是一个割容量为 C 的无环网络，那么在存在窃听对手且最多可以窃听 G 中的 z_I 条链路的情况下，其相干安全容量是 $C-z_I$。

在参考文献［9］中范式的基础上，Feldman 等人在参考文献［14］中对速率和域大小 q 之间可实现的权衡进行了研究。换句话说，他们证明了寻找矩阵 $\boldsymbol{T}$ 等价于寻找一个具有一定广义距离特性的纠错码，这里也称矩阵 $\boldsymbol{T}$ 为“过滤保密共享方案”。后者可通过与 Varshamov 在参考文献［79］中提出的相类似的随机线性构造方法来获得。基于上述关系，Feldman 等人在参考文献［14］中证明了对于任意的 $\sigma>1$，可以有效地构造一个矩阵 $\boldsymbol{T}$，该矩阵可以实现速率为 $C-\sigma z_I$ 且域大小为 $q=|E|^{\Omega(\frac{1}{\sigma-1})}$ 的安全通信。直观地看，所有具有适当广义距离的方案都能够将线性变换投射到 z_I 条链路上，这将使得窃听者只能获得一个均匀分布的随机变量。

在参考文献［61，62］中，Rouayheb 等人研究了参考文献［9］中的两步范式，提出了不同的方法，其集中于对内部网络编码方案的设计而不是对矩阵 $\boldsymbol{T}$ 的设计。也就是说，参考文献［61，62］将相干保密通信的任务与 Ozarow 和 Wyner 在其开创性工作参考文献［57，58］中提出的II型窃听信道相捆绑。后者通过陪集编码来实现前文提到的简单单跳网络中速率为 $C-z_I$ 的安全通信。参考文献［61，62］的作者观察到，这个不成熟的方案——即源节点使用参考文献［57，58］中的陪集编码技术对其信息预编码（通过 $\boldsymbol{T}$），然后再执行任一可行的网络编码方案——并不能确保安全通信。然而，参考文献［61，62］中的研究表明，如果该网络编码方案满足某些与所使用的特定陪集编码相关的条件，则确实可以实现速率为 $C-z_I$ 的安全通信。为了获得与给定的预编码方案 $\boldsymbol{T}$ 相匹配的网络编码，Rouayheb 等人对参考文献［29］中关于网络编码构造的确定性算法进行了修改（在没有窃听者的情况下），获得了当域大小近似于 $|E|^{z_I}$ 时，可实现最优速率为 $C-z_I$ 的可高效构造的安全码。为了降低域的大小，使其仅取决于 C、z_I 和多播连接中的终端节点数量 t，并与图 G 的大小相互独立，参考文献［61，62］的作者在

参考文献［42，43］中关键思想的基础上进行研究，并得到了一个域大小值 $\binom{2(C-z_I)^3t^2}{z_I-1}+t$，该值独立于$|V|$和$|E|$，但仍和其他网络参数保持指数关系。

Ngai 等人在参考文献［53］中对使用陪集编码的预编码进行了进一步的全面研究。Wei 等人在参考文献［83］中研究了线性分组码的广义汉明重量，受该工作的驱动，Ngai 等人定义了"网络广义汉明重量"的概念并提出了"网络 MDS"码。大体上说，这些概念将分组纠错码绑定到网络编码方案上，当和给定的网络编码方案结合使用时，它们足以表征允许安全通信的预编码方案 $\boldsymbol{T}$（反之亦然）。

Bhattad 等人在参考文献［7］中考虑了弱安全的概念，在其研究的场景中，窃听者确实有可能获得关于网络中多播消息的部分信息，但仅凭这部分信息不足以推断出源信息 X 中任一字符的确切值。例如，当传输由两个符号 a 和 b 组成的消息时，如果窃听者窃听到 $a+b$，则它能够获得关于（a，b）的部分信息，但是得不到 a 或 b 的精确值。与参考文献［9］中的范式类似，参考文献［7］中的研究工作表明，通过将源信息 X 与某矩阵 $\boldsymbol{T}$ 相乘，任何容量为 C 的网络编码方案都可以被转变成弱安全方案。对于该简化过程，参考文献［7］要求域的大小近似等于 $|E|^{\frac{k}{C-k}}$，其中 $k\leq z_I$ 是一个与窃听者从其所控链路中可能获得的信息总量相关的参数。对于其他一些相关的问题，参考文献［7］中还解决了在不进行任何基于 $\boldsymbol{T}$ 的预编码的情况下，随机线性网络码的完全/弱安全问题。与参考文献［26］相同，在该随机线性网络码中，支配编码方案的线性系数全都独立均匀地随机选取自域$\mathbb{F}$。此外，还假设窃听者的行为（即窃听者控制哪些链路）独立于对这些系数的随机选择。在上述设定下，可以得到域的大小和错误概率之间的权衡关系（对于完全安全和弱安全两种情况）。大体上说，如果在设计过程中错误概率 ϵ 是可容忍的，那么（在比较参考文献［9］中完全安全的界和参考文献［7］中弱安全的界时）域的大小将被乘以因子 $1/\epsilon$。

上述所有工作都集中于对无环网络的研究。在参考文献［9］中提出的分析技术的基础上，Jain 在参考文献［30］中研究了一般情况下（不一定是无环网络）的安全网络编码，即在窃听者可能会从给定的一组系统 $\mathscr{A}$ 中选择一个链路集合 $A\subseteq E$ 进行窃听的情况下对网络中的安全通信进行了一般性分析。当窃听对手最多可以控制 z_I 条链路时，$\mathscr{A}$ 刚好是由所有的最多包含 z_I 条链路的子集构成的一组系统。参考文献［30］中针对有环网络的情况加强了上述一般性分析，场景中单个源节点希望以单位速率与单个终端节点通信。参考文献［30］中从图 G 的拓扑和系统 $\mathscr{A}$ 的角度给出了能够以单位速率进行安全通信的充分必要条件。简单地说，安全通信的充分必要条件是：在对图 G 进行预处理并从中移除其信息不能到达终端节点的那些节点后，G 中存在一条未使用路径。换句话说，当将经过预处理的图看做是无向图时，存在一条对手看不见的从发送端到接收端的路径。在更高通信速率或更多终端节点的情况下，对安全通信的条件描述仍是开放性问题。

7.3.2 非相干情况

本节将集中讨论非相干场景下的通信问题，其中非相干指的是参与通信的各方事先

都不了解网络拓扑和所执行的网络编码操作。尽管有上述限制，但研究表明非相干场景下仍然可以实现与相干场景基本相同的性能。其工作集中在两条线上——采用随机信源编码的方案和采用确定信源编码的方案。

前一节中提到的 Feldman 等人在参考文献［14］中给出的构造方法属于第一种情况。他们随机地选择一个适当维度的矩阵，得到相应的线性滤波器，并将源节点的信息通过该滤波器。根据 Ho 等人在参考文献［26］中提出的分布式随机线性网络编码方案，网络中的内部节点在足够大的有限域上进行随机线性组合操作——事实上，统一这些操作可以产生一种非相干的分布式随机线性网络码，该码在网络中存在一个最多可以窃听 z_I 条链路的窃听者的情况下是完全安全的。这证明了对于最多窃听 z_I 条链路的窃听者，这些随机选择能够以很高的概率实现完全安全。此外，如果用 C 表示网络的最小割，则当编码方案使用的域的大小无限增大时，该方案可实现的通信速率能够无限接近于 $C-z_I$。还证明了当不要求安全通信的速率恰好等于 $C-z_I$ 时，一般所需的域的大小可以小得多。

定理 7-2[14]　对于任意的 $\sigma>1$ 且对于任意的域大小 $q\geqslant\max\{|E|^{\Omega(1/(\sigma-1))},|S|\}$，(其中 S 表示终端节点集合)，存在一个可行的速率为 $C-\sigma z_I$ 的线性网络编码方案，对于能够窃听 z_I 条链路的窃听者，该方案是完全安全的。此外，使用随机选择的源滤波器可以以很高的概率实现该性能。

它还证明了与此相反的是，如果期望的通信速率精确地等于 $C-z_I$，那么所需的域的大小最小应为 $|E|^{\Omega(\sqrt{z_I/\log z_I})}$。容量上的微小差距导致域大小在取值上的显著减少的原因是：将速率为 $C-\sigma z_I$ 的信息变成维度为 C 的信息所需的线性变换数量显著大于将速率为 $C-z_I$ 的信息变成维度为 C 的信息的情况。

受到 Rouayheb 和 Soljanin 推导的窃听网络公式的启发，在参考文献［78］中，Silva 和 Kschischang 在前人成果的基础上提出了一种可以叠加到非相干随机线性网络码（例如 Ho 等人在参考文献［26］中提出的方案）上的确定信源编码器。其提出了一种基于“最大秩距离”（MRD）码的陪集编码方案，该码既不用对所使用的网络码字施加任何约束条件，又不要求对所使用的网络码字有任何了解。也就是说，对于任意在多播场景中可行的线性网络码，均可以通过一个固定的外码来实现最大可能速率下的安全通信。特别指出的是，域的大小可以选择为多播所需的最小值。该方案的本质是对一个长度为 n 的分组使用向量线性外码，实际上也可以看做是扩展域 $\mathbb{F}_{q^n}$ 上的线性码。

定理 7-3[78]　通过域 $\mathbb{F}_q$ 上任一可行的线性网络码以及域 $\mathbb{F}_{q^n}$ 上一个固定的基于任一线性 MRD (C,z_I) 码的端到端陪集编码方案，可以实现速率为 $C-z_I$ 的完全安全通信。

7.4 干扰安全

本节中将研究网络中存在同时具有窃听和干扰能力的主动干扰者时，如何设计网络码才能实现可靠的错误检测和通信。将遵循概述小节中给出的大纲展开讨论。

7.4.1 相干情况

对于存在一个主动干扰者的相干场景，假设该干扰者能够干扰网络中的 z_O 条链路

并监测网络中的所有链路，要研究其中的纠错问题，首先研究该场景中的可靠通信速率。简要地说，我们证明了当 $C>2z_0$ 时，其可靠通信速率为 $C-2z_0$，否则为 0，即速率等于（$C-2z_0$）$^+$。从单跳单播网络出发，网络中源节点 s 希望通过 C 条（s，t）链路与终端节点 t 通信。假设链路传输的是大小为 q 的字符表 Σ 中的字符，且源节点需要发送 R 个这样的字符给 t。不难证明，在存在一个能够干扰 z_0 条链路的干扰对手的单跳单播网络中设计一种速率为 R 的可靠通信方案等价于设计一种可以纠正 z_0 个错误（即最小距离等于 $2z_0+1$）的［C，R］纠错码㊀。

令 Σ 的大小等于 q。对于定义在字符表 Σ 上的纠错码，假设其分组长度为 C，最小距离为 $2z_0+1$，则其速率 R（C，z_0，q）存在多个界。其中众所周知的是，对于 $\Delta=\log_q\left(\sum_{i=0}^{z_0}\binom{C}{i}(q-1)^i\right)$，有 $R(C,z_0,q)\leqslant C-\Delta$（参见参考文献［48］），该界又被称为球填充（Sphere Packing）界或汉明界。可以证明当 q 趋于无穷时，该界将趋近于 $C-z_0$。对于各种类型的错误——随机型或攻击型，该界均成立。此外，还有一个根据鸽笼原理推导出的 Singleton 界（参见参考文献［48］），该界表明 $R(C,z_0,q)\leqslant C-2z_0$。显然，当字符表的大小 q 足够大时，Singleton 界比球填充界更紧致。

R（C，z_0，q）的下界又是如何呢？诸如 Reed－Solomon 码等一些编码技术[48]表明，当速率等于 $C-2z_0$ 时，分组长度为 C 的纠错码可以纠正 z_0 个错误。其中 Gilbert[18]和 Varshamov[79]的研究工作与本章最为相关，他们证明了对于 $\Delta=\log_q\left(\sum_{i=0}^{2z_0}\binom{C}{i}(q-1)^i\right)$，不等式 $R(C,z_0,q)\geqslant C-\Delta$ 成立。需要注意的是，该等式中求和运算的范围是从 0 到 $2z_0$（Hamming 界中为 0 到 z_0）。根据上述讨论可知，当 q 趋于无穷时，Singleton 界是紧界，且对应于存在干扰者的单跳单播网络的容量。

那么，在更为复杂的网络中上述结论是否成立呢？Cai 和 Yeung 在参考文献［10，87］中研究了这个自然而有趣的问题，并给出了肯定的回答。参考文献［10，87］中作者在相干网络编码场景下对 Hamming 界、Singleton 界和 Gilbert－Varshamov 界进行了模拟。而且他们还证明了网络的 Singleton 型界和 q 值较大时的 Varshamov 型界是等价的。其分析的关键点在于理解网络的最小割集上传输的信息的组合性质，该割集将网络中的源节点和终端节点划分为两个分离的部分㊁。下面将给出关于参考文献［10，87］中研究成果的概述㊂。

㊀ 通过类似的论证，观察到当且仅当码的最小距离大于等于 z_0+1 时，该方案才能检测出能够干扰 z_0 条链路的干扰对手注入网络中的错误信息，也就是说能够检测出对手注入的错误信息的最大传输速率为 $C-z_0$。这里决定性的一点是，假设相干场景中的各方都知道所使用的（网络）码。对于干扰对手未能掌握网络中所有信息的情况，参考文献［25］中的工作证明了只要传输速率严格小于 C 就可以检测出对手注入的错误信息。

㊁ 参考文献［2］中给出了所需域大小的紧界，参考文献［3］中作者证明了如果略微降低对传输速率的要求，则可以显著减少相应域的大小——该结论与 Feldman 等人在参考文献［14］中对于窃听安全问题得出的结论相似。此外，参考文献［88］中还给出了能够在受到攻击的网络中纠错的高复杂度算法。

㊂ 在参考文献［74］中作者用“矩阵信道”的语言对上述分析（以及后文将介绍的参考文献［10，87］中的后续研究成果）进行了翻译，详情将在 7.4.2 节中讨论。

给定任意一个容量为 C 的（无差错）网络 $G=(V,\ E)$。令 A 和 B 表示 V 的一个划分，令割（A，B）表示起始节点属于 A 且终止节点属于 B 的所有链路的集合。为了模拟网络的 Hamming 界，参考文献［87］中研究了割集（A，B）上传输的信息，确切地说是源信息 X 和割集上传输的信息 $Z^m=Z_1,\ \cdots,\ Z_m$ 之间的映射关系，其中 $m=|\mathrm{cut}(A,\ B)|$。大体上说，如果 G 中不存在从 B 到 A 的直接链路，那么 Z^m 就必然是一个［m，$2z_0+1$］纠错码，这是因为此时终端 t 的解码仅仅是对 Z^m 的一个函数运算。实际上，如果 Z^m 的最小距离小于 $2z_0+1$，那么一个能够干扰割集（A，B）中的 z_0 条链路的恶意干扰者可能会导致 t 的解码错误。需要注意的是，上述简化关系仅在 B 与 A 之间没有直接链路时才存在，否则割集（A，B）中给定链路上的错误可能会影响其中的其他链路（对于标准纠错码模型，这些影响不会发生）。一旦在网络通信和纠错码之间建立了上述简化关系，则 Hamming 型界和 Singleton 型界有如下关系。

定理 7-4（网络 Hamming 界）　令 G 表示一个割容量为 C 的（无差错）无环网络，G 中的每条链路可以携带大小为 q 的字符表 Σ 中的一个字符。那么当 G 中最多有 z_0 条链路被干扰时，网络的相干容量不大于 $C-\Delta$，其中：

$$\Delta=\log_q\left(\sum_{i=0}^{z_0}\binom{C}{i}(q-1)^i\right)$$

固定 C 和 z_0，当域的大小 q 趋近于∞时，该界将趋近于 $C-z_0$。

对于经典纠错码，在网络中存在攻击型错误且 q 较大的情况下，网络容量的一个强约束可以通过对 Singleton 界进行网络模拟来获得。

定理 7-5（网络 Singleton 界）　令 G 表示一个割容量为 C 的（无差错）无环网络，那么，当 G 中最多有 z_0 条链路被干扰时，网络的相干容量不大于 $(C-2z_0)^+$。

下面讨论当网络中存在一个最多可以干扰 z_0 条链路的对手时，网络相干容量的下界。参考文献［10］中推导了网络通信背景下的 Gilbert - Varshamov 界。众所周知，在无差错相干场景中，使用诸如随机构造生成的线性网络码等方案可以成功地传输由不同消息组成的集合 Σ^C。参考文献［10］的主要思想是在使用上述网络码时准确地构造一个消息子集 $W\subset\Sigma^C$，该子集具有以下性质：无论对手选择哪个错误图样，终端都能够正确地区分出传输的信息 $w\in W$。也就是说，对于 Σ^C 中的两个码字 x 和 x'，如果存在两个错误图样 e 和 e'，其满足：当发送 x 且对手选择错误图样 e 时终端节点接收到的信息与发送 x' 且对手选择错误图样 e' 时接收到的信息相同，则认为 x 和 x' 是不可分的。参考文献［10］中的研究目标是寻找一个大的子集 W，其中任意 $w\in W$ 和 $w'\neq w\in W$ 都是可分的。上述分析的一个难题是对于给定的 $x\in\Sigma^C$，精确地寻找由所有与 x 不可分的码字 x' 组成的子集 V_x。通过界定 V_x 的大小 V 并依据 Gilbert 在参考文献［18］中提出的技术，可以得到大小为 q^C/V 的集合 W。此外，通过 Varshamov 型方法可以用 q^{2z_0} 界定 V，并得到大小为 q^{C-2z_0} 的线性 W。

定理 7-6（网络 Gilbert - Varshamov 界）　令 G 表示一个割容量为 C 的（无差错）无环网络，G 中的每条链路可以携带大小为 q 的字符表 Σ 中的一个字符。当 q 足够大且 G 中最多有 z_0 条链路被干扰时，网络的相干容量不小于 $(C-2z_0)^+$。

推论 7-1（相干容量）　令 G 表示一个割容量为 C 的（无差错）无环网络。那么，当 G 中最多有 z_0 条链路被干扰时，网络的相干容量等于 $(C-2z_0)^+$。

7.4.2　非相干情况

下面研究当非相干网络中隐藏着一个能够干扰 z_0 条链路的主动干扰者时，网络中的可靠通信速率问题。对于上述非相干场景，网络中的节点事先既不知道网络拓扑又不知道网络码字。研究表明，即便如此仍然可以实现与相干场景中相同的速率 $(C-2z_0)^+$。实际上，网络的内部节点可以忽视干扰对手的出现而仅执行某一预先“良好”设计的网络编码操作（如确定性多播网络编码或分布式随机网络编码）。所有的复杂度都源于编码器和解码器的操作，且该计算复杂度可以用网络参数的多项式形式表示。

实现上述性能的关键点在于以下观测结果。正如 7.2 节中提到的，当网络进行线性网络编码时源节点信息 $\boldsymbol{X}$、对手注入的伪造信息 $\boldsymbol{Z}$ 以及接收者收到的信息 $\boldsymbol{Y}$ 之间的关系可以表示为

$$\boldsymbol{Y}=\boldsymbol{TX}+\boldsymbol{T'Z} \tag{7-4}$$

$\boldsymbol{X}$ 和 $\boldsymbol{Y}$ 之间的上述关系也称为（线性）运算信道。

在参考文献［38，39］中作者提出了如下见解。令 $\mathscr{X}$、$\mathscr{Y}$ 和 $\mathscr{Z}$ 分别表示矩阵 $\boldsymbol{X}$、$\boldsymbol{Y}$ 和 $\boldsymbol{Z}$ 的行空间。那么式（7-4）表示向量空间 $\mathscr{Y}$ 恰好等于向量空间 $\mathscr{X}$ 和 $\mathscr{Z}$ 的直接相加，即包含 $\mathscr{X}$ 和 $\mathscr{Z}$ 的最小向量空间。然后在 $\mathbb{F}_q^n$ 的所有子空间的集合上定义了一个度量子空间 $d_S(.,.)$。对于 $\mathbb{F}_q^n$ 的任意子空间 $\mathscr{U}$ 和 $\mathscr{V}$，有：

$$d_S(\mathscr{U},\mathscr{V})=\mathbf{dim}\mathscr{U}+\mathbf{dim}\mathscr{V}-2\mathbf{dim}(\mathscr{U}\cap\mathscr{V})$$

可以看出，$d_S(.,.)$ 满足三角不等式，故上述定义确实可以表示一个度量空间。

从以上定义中可以找到在运算信道中设计“良好”码字的策略，该策略紧密平行于经典的代数码设计（如 **Reed－Solomon** 码）。参与通信的各方将预先选择一个由 $\mathbb{F}_q^n$ 的子空间组成的码本，且满足：每对子空间之间的子空间距离不小于 $4z_0+1$。在参考文献［38，39］中作者证明了当码本包含不少于 $q^{(C-2z_0)(n-C)}$ 个元素时，找到满足上述条件的码本是可能的[⊖]。对于长度为 n 的数据包和足够大的域 q，该值趋近于 q^{C-2z_0}。

接着，解码器进行如下操作——它从码本中寻找与观测到的空间 $\mathscr{Y}$ 之间的子空间距离最小的码字。由于干扰对手最多能够控制 z_0 条链路，也就是说 $\mathscr{Z}$ 的维度最大为 z_0，这将保证上述解码算法正确工作。

参考文献［38，39］的作者证明了通过基于线性化多项式的编码方案进行上述编码和解码的计算是高效的，该线性化多项式指的是对经典代数编码理论中 **Reed－Solomon** 码的模拟。接着他们在参考文献［71，73，77］中使用 **Gabidulin** 提出的秩度量解

⊖ 这个最优值是 $4z_0+1$ 而不是更为“直观”的根据经典编码理论得到的 $2z_0+1$ 的原因是：在最坏的情况下，对手注入的每个数据包都可以将 $\boldsymbol{TX}$ 的行空间维度减 1，同时添加一个 $\boldsymbol{TX'}$ 的行空间中的向量到 $\boldsymbol{TX}$ 上，其中 $\boldsymbol{X'}\neq\boldsymbol{X}$。因此每个由对手注入的数据包对子空间距离的影响最大可以到 2。而参考文献［74］中定义的另一种度量——注入度量——则不需要这额外的两倍。

码算法[17]展示了能够解码上述编码的另一种方法。

综合上述结果，可以得出以下定理：

定理 7-7（攻击型错误下的非相干容量）　当网络 G 中存在最多可以干扰 z_0 条链路的对手时，网络的非相干容量是$(C-2z_0)^+$。使用计算复杂度为 $O(C^2n)$ 的编码方案可以实现该容量。

在非相干场景中，对攻击型网络错误的检测问题要比纠错问题简单许多。依照参考文献［25］中提出的方案，源节点对它包含的每个数据包都添加一个散列。参考文献［25］的作者证明了只要从源节点到目的节点存在至少一条未损坏的路径，就可以通过一个低复杂度方案以较高的概率检测出对手施加的任何错误。

相关的工作还研究了链路上发生的是随机型错误而不是攻击型错误的情况。此时矩阵 $\boldsymbol{Z}$ 可以看做是均匀随机地选取自 $z_0\times n$ 个矩阵的集合，而不是对手以最小化发送者和接收者之间的通信速率为目的而刻意选择而得。

因此，原则上认为在随机型错误下可以实现大于攻击型错误时的传输速率。参考文献［50］中的工作以及参考文献［76］中的后续工作证明，事实确实如此。参考文献［76］中给出的证明过程简洁明了，下面简单地介绍其主要思想。

假设变换矩阵 $\boldsymbol{T}$ 是可逆的（注：在参考文献［26］中设计的随机网络码下，该假设成立的概率很高）。那么，式（7-4）可以写成如下形式：

$$\boldsymbol{Y}=\boldsymbol{T}(\boldsymbol{X}+\boldsymbol{T}^{-1}\boldsymbol{T}'\boldsymbol{Z}) \tag{7-5}$$

这里可以假设 $\boldsymbol{T}^{-1}\boldsymbol{T}'\boldsymbol{Z}$ 在该维度的所有矩阵组成的集合中均匀分布。那么此时的码字构造与参考文献［26］中的随机码字构造十分相似，唯一不同的是，需要用全零行和全零列来填充 $\boldsymbol{X}$。更确切地说，$\boldsymbol{X}$ 的前 z_0 行和前 z_0 列都被设为 0，剩下的$(C-z_0)\times(n-z_0)$子矩阵由一个$(C-z_0)\times(C-z_0)$单位子矩阵和一个$(C-z_0)\times(n-C)$载荷矩阵 $\boldsymbol{U}$ 组成，表示如下：

$$\begin{bmatrix}\boldsymbol{0}_{z_0\times z_0} & \boldsymbol{0}_{C\times z_0} & \boldsymbol{0}_{n\times C}\\ \boldsymbol{0}_{(C-z_0)\times z_0} & \boldsymbol{I}_{(C-z_0)\times(C-z_0)} & \boldsymbol{U}\end{bmatrix} \tag{7-6}$$

参考文献［76］中的研究表明，如果矩阵 $\boldsymbol{Y}$ 左侧顶部的 $z_0\times z_0$ 子矩阵满秩，则可通过 **Gauss－Jordan** 消元法得到接收矩阵 $\boldsymbol{Y}$ 的如下行简化阶梯形式：

$$\begin{bmatrix}\boldsymbol{Z}_1 & \boldsymbol{0}_{C\times z_0} & \boldsymbol{Z}_2\\ \boldsymbol{0}_{(C-z_0)\times z_0} & \boldsymbol{TI}_{(C-z_0)\times(C-z_0)} & \boldsymbol{TU}\end{bmatrix} \tag{7-7}$$

对于这种情况，由于假设 $\boldsymbol{T}$ 是可逆的，因此从式（7-7）的后 $C-z_0$ 行中可以重建信息载荷 $\boldsymbol{U}$。最后剩下的一步是证明对于 $\boldsymbol{Z}$，$\boldsymbol{Y}$ 确实能够以较高的概率满足之前假设的秩约束。参考文献［76］中的研究表明，当 q 或 n 较大时，$\boldsymbol{Y}$ 满足该秩约束的概率不小于 $1-o\ (1/q^{n-2z_0})$。

综合上述，可以得出以下定理[⊖]。

⊖ 参考文献[88]中也独立地提出了对于随机错误的一个类似的结论和算法。

定理 7-8(随机型错误下的非相干容量) 令 G 表示一个割容量为 C 的(无差错)无环网络,当 G 中最多有 z_0 条链路被注入随机数据包时,网络的非相干容量等于$(C-z_0)$的概率不小于$1-o\ (1/q^{n-2z_0})$。使用解码复杂度为$\mathcal{O}(C^2n)$的编码方案可以实现该容量。

最后讨论在非相干网络中可有效地纠错的一个替代方案,该方案在参考文献[28,29]中提出,平行于参考文献[38,39]中的工作。虽然当域的大小 q 或数据包的长度 n 趋于无穷时,二者可实现的速率渐近等价,但参考文献[27,28]中所提方案对参数的要求更严格,它所需的 q 和 n 更大,且其计算复杂度为 $\Theta\ (n^3)$,而不是参考文献[38,39]中的$\mathcal{O}\ (C_n^2)$。

但参考文献[27,28]中的证明技术有一个优点:它使用了计算效率很高的"线性列表解码"。给定对错误图样集的约束 z_0,如果解码器总是能够输出一个最大为 l 的列表,并确保该列表包含所传输的码字,就说该编码在速率 $R(C,z_0,q)$下 l 列可解。此外,如果该列表可以以$\mathbb{F}_q^n$上的子空间的仿射密码形式表示,即存在某个固定的向量 $\boldsymbol{v}$ 和某个固定的包含 l 个元素的子空间 $\mathscr{L}$,使得列表中的每个向量都可以表示为 $\boldsymbol{v}+\mathscr{L}$ 的形式,那么就说该编码在该速率下线性列表可解。

定理 7-9 (线性列表解码) 当 G 中最多有 z_0 条链路被干扰时,存在速率为$(C-z_0)$且线性 q^{C^2}列可解的编码,该编码的计算复杂度为$\mathcal{O}(C^3n)$。

上述定理的基本思路如下。编码器选择一个包含 q^{C-z_0}个矩阵 $\boldsymbol{X}$ 的码本,其中每个矩阵的秩都是 $C-z_0$。考虑到 $\boldsymbol{Z}$ 的秩最大为 z_0,因此由 $\boldsymbol{X}$ 和 $\boldsymbol{Z}$ 的行向量组成的矩阵的秩最大是 C——后文中不失一般性地假设该秩等于 C (如果不为 C,则对于更小的秩,也有相似的结论成立)。

解码器从 $\boldsymbol{Y}$ 中选择 C 个线性独立的列,用 $\boldsymbol{Y}^s$ 表示相应的矩阵,并用 $\boldsymbol{X}^s$ 和 $\boldsymbol{Z}^s$ 分别表示矩阵 $\boldsymbol{X}$ 和 $\boldsymbol{Z}$ 中与 $\boldsymbol{Y}^s$ 对应的列,根据公(7-4)可知 $\boldsymbol{Y}^s=[\boldsymbol{T}\mid\boldsymbol{T}']\begin{bmatrix}\boldsymbol{X}^s\\\boldsymbol{Z}^s\end{bmatrix}$。此外,由于 $\boldsymbol{Y}^s$ 可以看做是 $\boldsymbol{Y}$ 矩阵的列向量中的一组基,故存在一个矩阵 $\boldsymbol{F}$,满足 $\boldsymbol{Y}=\boldsymbol{Y}^s\boldsymbol{F}$。解码器可以通过计算 $(\boldsymbol{Y}^s)^{-1}\boldsymbol{Y}$ 来获得 $\boldsymbol{F}$。因此还可以将 $\boldsymbol{Y}$ 表示为

$$\boldsymbol{Y}=[\boldsymbol{T}\mid\boldsymbol{T}']\begin{bmatrix}\boldsymbol{X}^s\boldsymbol{F}\\\boldsymbol{Z}^s\boldsymbol{F}\end{bmatrix}\tag{7-8}$$

比较式(7-4)和式(7-8),并再次使用$[\boldsymbol{T}\mid\boldsymbol{T}']$(以较大的概率)可逆这一假设,可以得到:

$$\boldsymbol{X}=\boldsymbol{X}^s\boldsymbol{F}\tag{7-9}$$

$$\boldsymbol{Z}=\boldsymbol{Z}^s\boldsymbol{F}\tag{7-10}$$

特别指出的是,式(7-9)中给出了关于 $\boldsymbol{X}$ 的一组线性关系,解码器可将其用于列表解码方案。$\boldsymbol{X}^s$ 中的变量数量是 C^2,从而矩阵 $\boldsymbol{X}^s$ 中的元素在$\mathbb{F}_q$上张成一个维度为 C^2 的向量空间。接收端 Bob 上的列表正是相应地由 $\boldsymbol{X}^s\boldsymbol{F}$ 张成的 C^2 维向量空间 $\mathscr{L}$。

上述列表解码结论可以应用于多种场景。例如在参考文献[27,28]中,该结论被用于非相干网络纠错码的第一个阶段——首先,解码器使用该结论产生一个包含源信息

X 的仿射子空间；其次，解码器使用施加于码本的额外约束来完善该列表，该约束可以看做是码设计的一部分。由于该列表是仿射的，虽然它的规模较大（大小为 q^{C^2}），但上述完善过程仍能够以较高的计算效率实现。下节中会看到对该列表解码结论的另一个应用：密码协议。

注意：“包消失”错误是一种特殊的错误类型，正如若干研究人员观察到的（参见参考文献［38］），z_0 个数据包消失对应的速率损耗最大为 z_0，不论这些消失错误是随机产生的还是攻击导致的。与此形成对比的是，当上述错误是攻击型错误时，相应的速率损耗是 $2z_0$。因此，在 z_0 个消失错误下可实现的最大速率是 $C-z_0$。

7.4.3　加密场景

本节中将考虑网络中存在计算能力有限的对抗性干扰者的情况。应用一定的密码原语，可以防范这类干扰者。研究这一情况时，假设某些计算任务（例如离散对数或因子分解）是棘手的，并基于这些假设设计可行的通信方案。研究表明，当网络中存在一个最多能够损坏 z_0 条链路且计算能力有限的对手时，速率为 $C-z_0$ 的通信是可实现的。注意这对于 7.4.1 节和 7.4.2 节中在干扰者具有无限计算能力的条件下给出的速率 $C-2z_0$ 是一个提升。还需注意的是，该速率是可能实现的最大值。由于以下描述的大多方案的行为都是先检测对抗性攻击，再丢弃错误的数据包，故网络错误检测可以看做是这些方案的一个直接副产品。

大体上说，调查的工作可以分为两类：网络内认证和端到端认证。对于网络内认证的情况，需要设计某些认证机制，网络内的节点可以通过这些机制识别被干扰者损坏的消息数据包。一旦发现这些错误数据包，网络的中间节点就会将它们丢弃。这把存在干扰者的通信变成了无干扰者的通信——只是网络中的一些链路不能够传输信息。而对于后者，采用标准的随机线性网络编码方案（例如参考文献［26］）就可以进行可靠通信。在上述研究工作中，还存在一些挑战。例如设计有效的签名方案，这些方案一方面在线性编码操作下必须是封闭的（此种签名方案也被称为是同态的[4-6,12,31,49,60]），另一方面不需要能够在网络内部节点之间进行密钥分发的复杂基础设施。网络内认证可以保证速率为 $C-z_0$ 的通信，而且在许多情况下还可以实现更高的传输速率（取决于干扰者控制的具体链路）。

在端到端认证的情况下，网络内部的节点将忽视通信中存在对抗性干扰者这一事实，并采用没有干扰者的情况下普遍使用的标准编码协议。通过在源节点上的增强编码以及在终端节点上的专门解码来防范网络中的干扰者。相比于网络内认证方案，端到端方案在编码管理方面有着明显的优势，并可以保证与网络内认证方案相同的速率 $C-z_0$。但是，当基于某些“具体实例”对二者进行比较时，端到端认证实现的速率经常会低于网络内认证方案（干扰者的实际位置在这里起到了作用——端到端认证方案假设的是对手以最坏的方式放置自己的位置，这可能会过于悲观）。

1. 网络内认证

如果对于 $x=\sum_i x_i$，哈希函数 h 满足 $h(x)=\sum_i h(x_i)$，则称 h 是同态的。同态

哈希函数自然地适合于 Ho 等人在参考文献［26］中提出的随机（非相干）网络编码方案。节点接收到信息 y_e 和系数 $\{\alpha_i\}$（对于无差错场景应满足 $y_e = \sum \alpha_i x_i$，其中 x_i 是源信息）后，通过检查等式 $h(y_e) = \sum \alpha_i h(x_i)$ 是否成立来认证接收到的信息。这里假设网络的中间节点已知本地信息 $h(x_i)$。实际上，如果 h 是同态的且给定一个 y，要计算满足 $h(x) = y$ 的 x 是十分困难的，那么当 y_e 被损坏时，它不能通过认证的概率很大。

给定以上框架，自然地，需要研究上述本地信息 $h(x_i)$ 的一些需求。在 Krohn 等人以及 Gkantsidis 和 Rodriguez[21] 的工作中，假设源信息 $h(x_i)$ 的哈希函数能够被可靠地传输给网络的中间节点（如果对手可以伪造该信息，则它就可以注入能够通过中间节点认证过程的伪造信息）。因此，该方案需要一个集中且可信的授权者来提供这些哈希函数。参考文献［21，41］中提出的通信方案的安全性建立在求解离散对数问题十分困难的基础上。

认证所使用的哈希值的分发需要可靠信道，而在 Charles 等人[11]、Zhao 等人[13] 和 Boneh 等人[8] 的工作中，使用公钥加密系统来避免这一情形。参考文献［11］中提出的通信方案是建立在椭圆曲线上的离散对数问题和 co－Diffie－Hellman 计算问题的困难性的基础上。Zhao 等人在参考文献［13］中给出了一个基于线性子空间认证的方案，如果伪造信息 v 不在源信息张成的空间 V 中，则使用该方案可以避免对抗性干扰者将该信息注入网络。该方案完全依赖于求解离散对数问题的困难性。最后，参考文献［8］中给出了两个基于参考文献［13］中提出的线性子空间认证范式的方案。Boneh 等人在参考文献［8］中的研究表明，对于该认证范式，这两个方案的公钥大小可以说是最优的。参考文献［8］中的第一个方案基于 Diffie－Hellman 计算假设并且是同态的，第二个方案基于离散对数问题并且是参考文献［41］和［13］中方案的一个非同态变体。

定理 7-10（网络内认证） 令 G 表示一个割容量为 C 的（无差错）无环网络。当 G 中存在一个最多能够干扰 z_0 条链路且计算能力有限的对手时，使用网络内认证方案的网络容量是 $C - z_0$。

2. 端到端认证

与以上工作相似，参考文献［54］中 Nutman 和 Langberg 也考虑了对 Ho 等人在参考文献［26］中提出的（非相干）网络编码方案的加强。然而在参考文献［54］提出的通信方案中，网络的内部节点遵循与参考文献［26］中完全相同的协议——从而忽视对抗性干扰者的出现。唯一的不同之处在于源节点和终端节点的编码和解码过程。

更确切地说，参考文献［54］中的协议建立在 Jaggi 等人在参考文献［27，28］中提出的非相干方案的基础上（后者又建立在参考文献［26］的基础上），它具有以下总体架构。当网络中存在一个控制 $C - z_0$ 条链路且不受限制的干扰者（具有无限的计算能力）时，参考文献［27，28］中给出了一个速率为 $C - z_0$ 的非相干通信方案。对于 7.4.1 节讨论的情况，速率 $C - z_0$ 是不可能实现的，它只能在额外的假设条件下得到。实际上，参考文献［27，28］中额外地假设源节点和终端节点共享一个低速率的边信道，通过该信道，它们可以进行简短的（不为对抗性干扰者所知的）保密通信。参考

文献［27，28］中的分析基于以下观察：当网络中存在一个控制 z_0 条链路的干扰者时，终端节点通过列表解码（与其相对的是单一解码）可实现速率为 $C-z_0$ 的传输（见定理7-9)。终端节点一旦得到上述列表，就可以根据接收到的保密边信息从列表中选择正确的元素。对边信息的保密十分重要，这是为了避免干扰者施加针对性的错误使得终端节点的某些列表不能使用边信息消除歧义。

基于参考文献［27，28］中的列表解码结论（或其他任意列表解码方案，如参考文献［47］），参考文献［54］中进行了以下修正。与参考文献［27，28］中通过一条边信道（当前模型中没有给出）传输边信息不同，参考文献［54］中提出使用任一公钥加密方案（不一定是同态的）对边信息加密，并在网络上传输已加密的边信息。假设干扰者不能破解该加密方案，则可保证边信息仍是保密的，但还需将该边信息可靠地传输给终端节点。为了实现这一目标，参考文献［54］中使用任一7.4.2节提到的编码方案对加密边信息编码。根据边信息速率较低的这一事实，在编码边信息和编码源信息之间共享时间可以获得速率 $C-z_0$。需要注意的是，参考文献［54］中指出，要保证边信息的可靠传输需要满足 $C>2z_0$，并证明了这一条件的必要性（在一定假设前提下）。

定理7-11（端到端认证） 令 G 表示一个割容量为 C 的（无差错）无环网络。当 G 中存在一个最多能够干扰 z_0 条链路且计算能力有限的对手时，使用端到端认证方案的网络容量是 $C-z_0$，其中 C 和 z_0 需要满足 $C>2z_0$。

7.5 存在窃听和干扰对手时的保密传输

7.5.1 相干情况

本节中考虑窃听和干扰之间的相互影响。正如在7.3节中看到的，至少要损耗大小为 z_1 的速率，才能保护某消息不被能够监听 z_1 条链路的窃听者窃听。同时还证明了在该速率损耗下实现速率为 $C-z_1$ 的保密通信的分布式低复杂度方案是存在的。本质上说，这些方案的实现均是基于对速率为 z_1 的随机消息与速率为 $C-z_1$ 的源消息的线性混合。因此，可以将这些方案看做是结合了网络编码的一次一密设计。

接下来在7.4节中看到，当网络中隐藏着一个能够监听所有传输并干扰 z_0 条链路对抗性干扰者时，源节点与目的节点之间的信息传输速率将降低到 $C-2z_0$，且存在实现该速率的分布式低复杂度方案。可以认为这些方案的作用是将一条容量为 C 的差错运算信道转变为一条容量为 $C-2z_0$ 的无差错运算信道。

对于对手只能监听网络中的 z_1 个传输并干扰 z_0 条链路的场景，我们很自然地会问，该场景中最大可实现的保密且可靠通信的速率是多少？参考文献［52］中的研究表明，在零错误和单字符编码的情况下，该速率是前文给出的两个速率界的“自然”组合，即总体速率为 $C-2z_0-z_1$。参考文献［52］中通过与前两节相似的用于界定速率的技术证明了这一点。参考文献［72］和［75］中对该界进行了基于零错误分组长度编码的扩

展⊖。此外，相干情况下满足上述速率界的算法在参考文献［51］（基于分组编码）和参考文献［52］（基于单字符编码）中给出。

本质上看，这些算法都是对前两节所给出算法的合并——它们首先构建一个将差错运算信道转变为速率为 $C-2z_O$ 的无差错运算信道的编码方案，然后在这个信道上覆盖一个防止对手窃听的“一次一密 + 网络编码”方案，从而将传输速率进一步降低为 $C-2z_O-z_I$，也就是总体速率。综上所述，可得到以下定理。

定理 7-12[51,52,72,75]　当网络中隐藏着一个可以窃听 z_I 条链路并干扰 z_O 条链路的对手时，保密信息的最大可靠（无差错）传输速率是 $C-2z_O-z_I$。

有趣的是，如果不要求无差错传输，而仅要求出现错误的概率较小（该概率随域的大小或分组长度的增大渐近于零），那么该上界将不再是 $C-2z_O-z_I$，而是 $C-z_O-z_I$。实际上，可以设计一些低复杂度编码来实现这个更高的速率，具体内容将在下节阐述。

7.5.2　非相干情况

在参考文献［72，75］中作者将 7.5.1 节给出的结果延伸为通用的编码设计。也就是说，对于任一线性变换的秩为 C 的线性网络码，参考文献［72，75］中给出了相应的端到端方案，该方案将网络码看做是一个运算信道，并可以实现与 7.5.1 节中的方案相同的速率为 $C-2z_O-z_I$ 的保密通信。上述构造建立在秩度量码的基础上——研究表明，这样的编码方案不仅在 7.4.2 节所述的纠错方面是良好的，而且在对源节点线性映射的保密性保持方面也有较好的表现。

参考文献［86］中的进一步研究表明，只要对手的干扰速率 z_O 和窃听速率 z_I 之和小于网络容量 C（即 $z_O+z_I<C$），就存在能够正确传输（出现错误的概率趋近于零）单个比特且不会泄露任何信息给对手的低计算复杂度的编码方案。将该方案与参考文献［27，28］中提出的“秘密共享”技术结合，可以设计允许最优源速率 $C-z_O+z_I$ 下的通信且能够防止对手获得通信信息的编码方案。其中，参考文献［27，28］中提出的“秘密共享”技术指的是：

定理 7-13[27,28]对于其中隐藏着一个最多可以干扰 z_O 条链路的对手的网络，如果源节点能够保密且可靠地将 ϵn（对于任意固定的 $\epsilon>0$）个比特传输给目的节点，那么实际上（$C-z_O-z_I$）n 个比特能够被保密且可靠地从源节点传输到目的节点。

定理 7-13 的基本原理如下：如果源节点基于其信息产生一个“较小的”保密线性哈希函数，并通过保密可靠信道将它发送给接收节点，那么，根据定理 7-9 给出的线性列表解码结论，接收节点能够以较大的概率从列表中提取出单一的元素。

那么剩下的问题仅仅是描述这个在网络中保密且可靠地共享一个比特数据（模拟一条可靠保密信道）的协议。参考文献［86］中使用以下简单“秩调制”协议来实现

⊖　实际上，参考文献［72］中证明了更具一般性的下界是 $C-2z_O-z_I-\rho$，其中 ρ 是消失错误（可以是对抗性的）的数量。

这一点。如果待共享的比特数据为0，则将源信息表示为一个秩为 $C-z_0-1$ 的矩阵；否则将它表示为一个秩为 C 的随机矩阵。解码器通过估算接收到的矩阵的秩进行解码。如果该秩等于 C，则解码器将保密比特判决为1，否则判决为0。

为了检验以上协议成功的概率，需分别检验其安全性和可靠性。对手最多窃听到 z_I 个传输，考虑到网络中的随机线性混合运算以及 $C-z_0-z_I>0$ 这一约束，可知对手窃听到的信息不足以让它区分出秩为 $C-z_0-1$ 的源信息和秩为 C 的源信息之间的区别，从而得以保证传输的安全性。可靠性的保证来源于以下两方面。首先，对手最多注入 z_0 个数据包，如果源信息为0，则源节点发送一个秩为 $C-z_0-1$ 的矩阵，那么接收节点收到的矩阵的秩仍然小于 C；反之，如果源信息是1，则源节点发送一个秩为 C 的随机矩阵，由于对手不知道这个矩阵的具体取值，因此它注入的数据包能够减少接收矩阵的秩的概率非常小。

7.6 其他方案

本节中总结了其他的一些与网络的保密可靠通信这一主题相关，但又不能归入前文章节中的研究工作。

- 参考文献［36，37，63］中研究了在链路容量不等的场景中的网络纠错问题（带反馈和不带反馈）——该情况下对可实现速率的完整描述仍然是一个开放性问题。
- Kosut 等人在参考文献［55，56］中考虑了网络中的节点而不是链路被对手控制的情况。同样，尚未有文献对该情况下的速率域进行完整描述。此外，参考文献［82］中也研究了在网络中存在不可信节点时使用网络编码实现可靠通信的问题。
- 参考文献［68－30，81，84］中从多路访问的角度讨论了网络纠错方案的变体。
- 参考文献［47］中提出了网络纠错码的非琐细列表解码算法，它是对［22］中考虑的点对点信道经典算法的模拟。
- 参考文献［59］中研究了分布式数据存储中的可靠性与保密性问题。
- 参考文献［15，16，19，23，65－67，85］中研究了寻找网络错误的实际位置这一问题。
- 参考文献［45，80］中提出了安全实用的网络编码方案（SPOC），允许在有计算能力有限并可能对网络上传输的所有通信进行窃听的对手情况下进行私密通信。在其核心部分，SPOC 运行的是随机线性编码方案的一种变体[26]，其中每个数据包的报头（含有编码系数）是加密的，且对于对手来说是未知的，同时数据包的数据部分（包含已通过网络编码的信息）透明发送。
- 在参考文献［33－35］中，作者研究了当无线网络中的对抗性节点可能会有恶意行为时的错误检测问题。通过代数监督方案，上游节点可以利用无线媒介的广播特性概率性地检测恶意行为。

7.7 总结

本章对在网络中存在被动和主动攻击者的情况下多播网络编码使用的编码方案进行了简单的概述。我们已经看到，当网络中存在一个可以控制 z_I 条链路的被动窃听者时，速率为 $C-z_I$ 的非相干保密通信是可实现的。对于相干通信和非相干通信，该速率都是可能实现的最大值。对于主动干扰者的情况，证明了当网络中的 z_O 条链路被干扰者控制时，速率为 $C-2z_O$ 的非相干可靠通信是可实现的。如果该干扰者的计算能力有限，则可以实现更高的速率 $C-z_O$。与之前相同，对于相干通信和非相干通信，这些速率都是可能实现的最大值。最后，当网络中存在能够干扰 z_O 条链路并窃听 z_I 条链路的对手时，$C-2z_O-z_I$ 是可实现的安全且可靠通信速率的一个紧界（当允许错误以较小的概率出现时，该值为 $C-z_O-z_I$）。本章中给出的算法技术覆盖了许多范式，并包含了组合数学、线性代数和编码理论等研究工具。本章解决了无环网络中的多播问题。在更具一般性的场景中（存在或不存在对手），对网络编码能力的理解仍是一个有趣的研究领域，并且在未来的几十年里一定会有所发展。

7.8 致谢

作者要特别感谢 Danilo Silva 在本章节的准备过程中给予的许多帮助性讨论。

参考文献

[1] R. Ahlswede, N. Cai, S. R. Li, and R. W. Yeung. Network informationflow. IEEE Transactions on Information Theory, 46 (5): 1204 - 1216, July 2000.

[2] H. Balli, X. Yan, and Z. Zhang. Error correction capability of random network error correction codes. In Proc. International Symposium on Information Theory, Sept. 2007.

[3] H. Balli, X. Yan, and Z. Zhang. On randomized linear network codes and their error correction capabilities. IEEE Transactions on Information Theory, 55 (7): 3148 - 3160, 2009.

[4] N. Baric and B. Pfitzmann. Collision - free accumulators and failstop signature schemes without trees. In Advances in Cryptology EUROCRYPT, 1997.

[5] M. Bellare and D. Micciancio. A new paradigm for collision - free hashing: Incrementality at reduced cost. In Advances in Cryptology EUROCRYPT, 1997.

[6] J. Benaloh and M. de Mare. One - way accumulators: A decentralized alternative to digital sinatures. In Advances in Cryptology EURO CRYPT, 1993.

[7] K. Bhattad and K. R. Narayanan. Weakly secure network coding. In proceedings of NetCod, 2005.

[8] D. Boneh, D. Freeman, J. Katz, and B. Waters. Signing a linear subspace: Signature schemes for network coding. In 12th International Conference on Practice and Theory in Public Key Cryptography, pages68 - 87, 2009.

[9] N. Cai and R. W. Yeung. Secure network coding. In Proceedings of International Symposium on Information Theory, Lausanne, Switzerland, June 2002.

[10] N. Cai and R. W. Yeung. Network error correction, part II: Lower bounds. Commun. Inf. Syst, 6 (1): 37 - 54, 2006.

[11] D. Charles, K. Jain, and K. Lauter. Signatures for network coding. In Proceedings of the Fortieth Annual Conference on Information Sciencesand Systems, Princeton, NJ, USA, 2006.

[12] D. Chaum, E. van Heijst, and B. Pfitzmann. Cryptographically strong undeniable signatures, unconditionally secure for the signer. In Advances in Cryptology CRYPTO, 1991.

[13] M. M'edard, F. Zhao, T. Kalker, and K. J. Han. Signatures for content distribution with network coding. In Proceedings of International Symposium on Information Theory (ISIT 2007), pages 556 - 560, 2007.

[14] J. Feldman, T. Malkin, C. Stein, and R. A. Servedio. On the capacity of secure network coding. In Proceedings of 42nd Annual Allerton Conference on Communication, Control, and Computing, Monticello, IL, 2004.

[15] C. Fragouli and A. Markopoulou. A network coding approach to network monitoring. In Proc. of the 43nd Allerton Conference, 2005.

[16] C. Fragouli, A. Markopoulou, and S. Diggavi. Topology inference using network coding. In Proc. of the 44nd Allerton Conference, 2006.

[17] E. M. Gabidulin. Theory of codes with maximum rank distance. Probl. Inform. Transm., 21 (1): 1 - 12, 1985.

[18] E. N. Gilbert. A comparison of signalling alphabets. Bell Systems Technical Journal, 31: 504 - 522, 1952.

[19] M. Gjoka, C. Fragouli, P. Sattari, and A. Markopoulou. Loss tomographyin general topologies with network coding. In Proc. ofIEEE Globecom, 2005.

[20] C. Gkantsidis and P. Rodriguez. Network coding for large scale content distribution. In Proceedings of IEEE Conference on Computer Communications (INFOCOM), Miami, March 2005.

[21] C. Gkantsidis and P. Rodriguez. Cooperative security for network coding file distribution. In Proceedings of IEEE Conference on Computer Communications (INFOCOM), pages 1 - 13, Barcelona, April 2006.

[22] V. Guruswami and M. Sudan. Improved decoding of Reed - Solomon and algebraicgeometric codes. IEEE Transactions on Information Theory, 45: 1757 - 1767, September 1999.

[23] L. Hailiang, H. Guangmin, Q. Feng, and Y. Zhihao. Network topology inference based on traceroute and tomography. In Proc. of International Conference on Communications and Mobile Computing, 2009.

[24] T. Ho., R. Kötter, M. M'edard, D. Karger, and M. Effros. The benefitsof coding over routing in a randomized setting. In IEEE International Symposium on Information Theory (ISIT), page 442, Yokohama, July 2003.

[25] T. Ho, B. Leong, R. Koetter, M. M'edard, M. Effros, and D. R. Karger. Byzantine modification detection in multicast networks using randomized network coding. IEEE Transactions on Information

Theory, 54 (6): 2798 - 2803, 2008.

[26] T. Ho, M. M'edard, R. Kötter, D. R. Karger, M. Effros, J. Shi, and B. Leong. A random linear network coding approach to multicast. IEEE Transactions on Information Theory, 52 (10): 4413 - 4430, 2006.

[27] S. Jaggi, M. Langberg, S. Katti, T. Ho, D. Katabi, and M. M'edard. Resilient network coding in the presence of Byzantine adversaries. In Proc. 26th IEEE Int. Conf. on Computer Commun., pages 616 - 624, Anchorage, AK, May 2007.

[28] S. Jaggi, M. Langberg, S. Katti, T. Ho, D. Katabi, M. M'edard, and M. Effros. Resilient network coding in the presence of Byzantine adversaries. IEEE Transactions on Information Theory, 54 (6): 2596 - 2603, June 2008.

[29] S. Jaggi, P. Sanders, P. A. Chou, M. Effros, S. Egner, K. Jain, and L. Tolhuizen. Polynomial time algorithms for multicast network codeconstruction. IEEE Transactions on Information Theory, 51 (6): 1973 - 1982, June 2005.

[30] K. Jain. Security based on network topology against the wiretapping attack. IEEE Wireless Communications, pages 68 - 71, Feb 2004.

[31] R. Johnson, D. Molnar, D. Song, and D. Wagner. Homomorphic signature schemes. In Progress in Cryptology CT RSA, 2002.

[32] S. Katti, H. Rahul, D. Katabi, W. Hu M. M'edard, and J. Crowcroft. XORs in the Air: Practical Wireless Network Coding. In ACM SIGCOMM, Pisa, Italy, 2006.

[33] M. Kim, M. M'edard, and J. Barros. A multi - hop multi - sourcealgebraic watchdog. In IEEE Information Theory Workshop (ITW), Dublin, Ireland, August 2010.

[34] M. Kim, M. M'edard, and J. Barros. Algebraic watchdog: Mitigating misbehavior in wireless network coding. Submitted to IEEE Journalon Selected Areas in Communications (JSAC), Advances in MilitaryNetworking and Communications, http: //arxiv. org/abs/1011. 3879, November 2010.

[35] M. Kim, M. M'edard, J. Barros, and R. Kötter. An algebraic watchdogfor wireless network coding. In IEEE International Symposium on Information Theory (ISIT), June 2009.

[36] S. Kim, T. Ho, M. Effros, and S. Avestimehr. Network error correction with unequal link capacities: Capacities and upper bound. Inpreparation for submission to the IEEE Transactions on Information Theory, 2009.

[37] S. Kim, T. Ho, M. Effros, and S. Avestimehr. New results on network error correction: Capacities and upper bound. In Proceedings of Information Theory and Applications Workshop, UCSD, San Diego, CA, 2010.

[38] R. Kötter and F. R. Kschischang. Coding for errors and erasures in random network coding. In Proc. IEEE Int. Symp. Information Theory, pages 791 - 795, Nice, France, June 24 - 29, 2007.

[39] R. Kötter and F. R. Kschischang. Coding for errors and erasures in random network coding. IEEE Transactions on Information Theory, 54 (8): 3579 - 3591, Aug. 2008.

[40] R. Kötter and M. M'edard. An algebraic approach to network coding. IEEE/ACM Transactions on Networking, 11 (5): 782 - 795, October2003.

[41] Maxwell N. Krohn, Michael J. Freedman, and David Mazires. Onthe – fly verification of rateless erasure codes for efficient content distribution. In Proceedings of the IEEE Symposium on Security and Privacy, pages 226 – 240, Oakland, California, 2004.

[42] M. Langberg, A. Sprintson, and J. Bruck. The encoding complexity of network coding. IEEE Transactions on Information Theory (a jointspecial issue with IEEE/ACM Transactions on Networking on Networking and Information Theory), 52 (6): 2386 – 2397, 2006.

[43] M. Langberg, A. Sprintson, and J. Bruck. Network coding: A computational perspective. IEEE Transactions on Information Theory, 55 (1): 147 – 157, 2009.

[44] S. – Y. R. Li, R. W. Yeung, and N. Cai. Linear network coding. IEEE Transactions on Information Theory, 49 (2): 371 – 381, 2003.

[45] L. Lima, J. P. Vilela, J. Barros, and M. M'edard. An informationtheoretic cryptanalysis of network coding – is protecting the code enough? In Proceeding of the International Symposium on Information Theory and its Applications (ISITA), 2008.

[46] D. S. Lun, M. M'edard, and R. K¨otter. Efficient operation of wireless packet networks using network coding. In International Workshop on Convergent Technologies (IWCT), Oulu, Finland, 2005.

[47] H. Mahdavifar and A. Vardy. Algebraic list – decoding on the operator channel. In Proc. of International Symposium on Information Theory (ISIT), Austin, TX, USA, June 13 – 19 2010.

[48] F. J. McWilliams and N. J. A. Sloane. The Theory of Error – Correcting Codes. North – Holland, 1977.

[49] S. Micali and R. Rivest. Transitive signature schemes. In Progress in Cryptology CT RSA, 2002.

[50] A. Montanari and R. Urbanke. Coding for network coding. Available online at http: //arxiv. org/abs/0711. 3935.

[51] C. K. Ngai and S. Yang. Deterministic secure error – correcting (SEC) network codes. In Proc. IEEE. Information Theory Workshop, pages 96 – 101, Bergen, Norway, June 24 – 29, 2007.

[52] C. – K. Ngai and R. W. Yeung. Secure error – correcting (SEC) net work codes. InWorkshop on Network Coding, Theory and Applications (NetCod), pages 791 – 795, Lausanne, Switzerland, June 2009.

[53] C. K. Ngai, R. W. Yeung, and Z. Zhang. Network Generalized Hamming Weight. In Proc. 2009 Workshop on Network Coding, Theory, and Applications, 2009.

[54] L. Nutman and M. Langberg. Adversarial models and resilient schemes for network coding. In Proceedings of International Symposium on Information Theory, pages 171 – 175, 2008.

[55] L. Tong, O. Kosut, and D. Tse. Nonlinear network coding is necessary to combat general byzantine attacks. In Proc. of 47th Annual Allerton Conference on Communication, Control, and Computing, October 2009.

[56] L. Tong, O. Kosut, and D. Tse. Polytope codes against adversaries in networks. In Proc. of International Symposium on Information Theory (ISIT), Austin, TX, USA, June 13 – 19, 2010.

[57] L. H. Ozarow and A. D. Wyner. The wire – tap channel II. Bell Syst. Tech. Journ., 63: 2135 – 2157, 1984.

[58] L. H. Ozarow and A. D. Wyner. Wire - tap channel II. In Proceedings of the EUROCRYPT 84 workshop on advances in cryptology: Theory and application of cryptographic techniques, pages 33 - 51, 1985.

[59] S. Pawar, S. El - Rouayheb, and K. Ramchandran. On secure distributed data storage under repair dynamics. In Proc. IEEE Int. Symp. Information Theory, June 13 - 19 2010.

[60] T. P. Pedersen. Non - interactive and information - theoretic secure verifiable secret sharing. In Advances in Cryptology CRYPTO, 1991.

[61] S. E. Rouayheb, E. Soljanin, and A. Sprintson. Secure network coding for wiretap networks of type II. Manuscript available on arXiv, org, 2009.

[62] S. E. Rouayheb and E. Y. Soljanin. On Wiretap Networks II. In proceedings of IEEE International Symposium on Information Theory, pages 551 - 555, 2007.

[63] M. Effros, S. Kim, T. Ho, and S. Avestimehr. Network error correction with unequal link capacities. In Proceedings of 47th Annual Allerton Conference on Communication, Control, and Computing, Monticello, IL, 2008.

[64] C. E. Shannon. Communication theory of secrecy systems. Bell System Technical Journal, 28 (4): 656 - 715, 1949.

[65] J. M. Siavoshani, C. Fragouli, and S. Diggavi. Subspace properties of randomized network coding. In Proc. of IEEE ITW, 2007.

[66] M. Jafari Siavoshani, C. Fragouli, and S. Diggavi. On locating byzantine attackers. In Network Coding Workshop: Theory and Applications, 2008.

[67] M. Jafari Siavoshani, C. Fragouli, S. Diggavi, and C. Gkantsidis. Bottleneck discovery and overlay management in network coded peerto - peer system. In Proc. of SIGCOMM Workshop on Internet Network Management, 2007.

[68] M. J. Siavoshani, C. Fragouli, and S. Diggavi. Noncoherent multisource network coding. In Proc. IEEE Int. Symp. Information Theory, pages 817 - 821, Toronto, Canada, July 6 - 11, 2008.

[69] M. J. Siavoshani, C. Fragouli, and S. Diggavi. Code construction for multiple sources network coding. In Proc. of the MobiHoc, 2009.

[70] M. J. Siavoshani, S. Mohajer, C. Fragouli, and S. Diggavi. On the capacity of noncoherent network coding. In Proc. of the IEEE International Symposium on Information Theory, 2009.

[71] D. Silva and F. R. Kschischang. Using rank - metric codes for error correction in random network coding. In Proc. IEEE Int. Symp. Information Theory, pages 796 - 800, Nice, France, June 24 - 29, 2007.

[72] D. Silva and F. R. Kschischang. Universal secure network coding via rank - metric codes. IEEE Transactions on Information Theory, 57 (2): 1124 - 1135, 2011.

[73] D. Silva and F. R. Kschischang. Fast encoding and decoding of Gabidulin codes. In Proc. IEEE Int. Symp. Information Theory, Seoul, Korea, June 2009.

[74] D. Silva and F. R. Kschischang. On metrics for error correction in network coding. IEEE Transactions on Information Theory, 55 (12): 5479 - 5490, 2009.

[75] D. Silva and F. R. Kschischang. Universal secure error control schemes for network coding. In

Proc. IEEE Int. Symp. Information Theory, Austin, TX, USA, June 13 - 19, 2010.

[76] D. Silva, F. R. Kschischang, and R. K¨otter. Capacity of random network coding under a probabilistic error model. In Proc. 24th Biennial Symposium on Communications, Kingston, ON, Canada, June 2008.

[77] D. Silva, F. R. Kschischang, and R. K¨otter. A rank - metric approach to error control in random network coding. IEEE Transactions on Information Theory, 54 (9): 3951 - 3967, 2008.

[78] Danilo Silva and Frank R. Kschischang. Universal weakly secure network coding. In Proc. Inform. Theory Workshop on Networking and Inform. Theory, pages 281 - 285, Volos, Greece, June 10 - 12, 2009.

[79] R. R. Varshamov. Estimate of the number of signals in error correcting codes. Dokl. Acad. Nauk, 117: 739 - 741, 1957.

[80] J. P. Vilela, L. Lima, and J. Barros. Lightweight security for network coding. In Proc. of the IEEE International Conference on Communications (ICC 2008), Beijing, China, May 2008.

[81] S. Vyetrenko, T. Ho, M. Effros, J. Kliewer, and E. Erez. Rate regions for coherent and noncoherent multisource network error correction. In Proc. of International Symposium on Information Theory (ISIT), Seoul, Korea, June 2009.

[82] D. Wang, D. Silva, and F. R. Kschischang. Robust network coding in the presence of untrusted nodes. IEEE Transactions on Information Theory, 56 (9): 4532 - 4538, 2010.

[83] V. K. Wei. Generalized HammingWeight for linear codes. IEEE Trans. Inform. Theory, 37 (5): 1412 - 1418, 1991.

[84] H. Yao, T. K. Dikaliotis, S. Jaggi, and T. Ho. Multiple access network informationflow and correction codes. In Proc. IEEE Information TheoryWorkshop, Dublin, Ireland, 2010.

[85] H. Yao, S. Jaggi, and M. Chen. Passive network tomography for erroneous networks: A network coding approach. Under submission to the IEEE Transactions on Information Theory, 2010.

[86] H. Yao, D. Silva, S. Jaggi, and M. Langberg. Network codes resilient to jamming and eavesdropping. In Proc. Workshop on Network Coding Theory and Applications, Toronto, Canada, June 9 - 11 2010.

[87] R. W. Yeung and N. Cai. Network error correction, part I: Basic concepts and upper bounds. Commun. Inf. Syst, 6 (1): 19 - 36, 2006.

[88] Z. Zhang. Linear network error correction codes in packet networks. IEEE Transactions on Information Theory, 54 (1): 209 - 218, January 2008.

第8章 网络编码和数据压缩

Mayank Bakshi 美国，加利福尼亚州，帕萨迪纳，加州理工学院，电子工程系

Michelle Effros 美国，加利福尼亚州，帕萨迪纳，加州理工学院，电子工程系

Tracey Ho 美国，加利福尼亚州，帕萨迪纳，加州理工学院，电子工程系

Muriel Medard 美国，马萨诸塞州，剑桥，麻省理工学院，电子工程系

摘要

本章考虑网络上通信统计相关源的问题，特别是对于在无环网络中的无噪链路上进行无损重构的情况。与信源独立的纯网络编码设定相反，该场景下的最优码设计不仅包括网络编码，还包括数据压缩。尽管可达速率范围的一些连续特性是已知的，并且根据割集理论可以获得一般的外界，但一般来说计算该问题的可达速率集仍是一个非常困难的任务。尽管这些割集界不是在所有的网络中都可以实现，但对于组播网络它们是可实现的，也就是那些所有汇聚节点均点播了所有源节点的网络。对于扩展的组播网络来说，这些割集界也是可实现的。在扩展的组播网络中，除了期望的源节点外，可能还有一些汇聚节点具有边信息。对于这两种情况，均可利用线性网络编码实现割集界。本章的最后，我们讨论了结合了网络编码和压缩的实用性方法。

关键字

分布式压缩、联合源－网络编码、无损、组播、网络信源编码、边信息。

8.1 引言

网络源编码问题和网络编码问题密切相关。这两种情形都是假设一个链路无噪且容量有限的网络。每个节点可以观测到一个或多个源，并有一个或多个通信需求。在最高层需要考虑的问题是：我们是否可以以一种能够满足所有节点的相应需求的方式来设计描述网络中的源的码字。

在网络源编码中，来自网络中不同节点的源一般假设为任意分布，即允许任意统计相关。源字母集可离散也可连续。需求既可以是网络中其他可用节点上的源[1, 2]，也可以是这些源的函数[3, 4]。与每个需求相关联的是重建保真度的要求等级。保真度要求的范围从零误码重建开始（源与重建之间存在不同的概率精确为零，示例请参阅参考文献［5，6］），到无损重建（源与重建之间存在不同的概率，可达到任意小但不为零，示例请参阅参考文献［1］），再到有损重建（即给定具体的失真测度，并限制最大失真，示例请参阅参考文献［7，8］）。关于网络源编码的主要文献致力于处理网络源编码问题的特例，比如修补网络拓扑结构、收集需求、分析可能满足给定需求的链路容量值的空间的全部或部分特性（示例请参阅参考文献［1，2，7－11］）。

在网络编码领域的最早的研究工作中，网络编码被定位为数据传输问题而非数据表示问题。结果就是，大多网络编码文献通常都将通过网络的源看做是有限字符集上的独立同分布随机变量。文献中考虑了无损（见参考文献 [12]）和零误码（见参考文献 [13, 14]）两种情形。尽管大多数的网络编码文献处理的都是独立同分布源，但也有一些文献考虑了相关源的一般情形（参见参考文献 [15]）。对于该情形，网络编码和无损网络源编码是等价的。大多网络编码文献考虑的都是一般的编码场景而非特定拓扑。组播和多源组播问题，即一个或多个发送设备将相同的信息发送给某个固定的接收机集合中的所有接收机，就是这种编码场景的示例（参见参考文献 [12, 15]）。在这些编码场景中，参考文献给出了对于拓扑任意但需求类型固定的网络的推导结果。

网络编码问题和网络源编码问题不仅在概念上相似，而且从某种程度上说它们不可分割。参考文献 [16] 和 [17] 分别给出了只有联合网络编码和网络源编码才能获得最佳端到端通信性能的通信场景的非组播和组播的示例。对于这些场景，分离这两种方案意味着，每个源只能在它的源节点上被压缩一次，然后就在网络中被传输，中间节点不知晓其统计特性，也不在网络中的其他地方被压缩。图 8-1a 给出了参考文献 [16] 中示例的一个实例。实例中源 X_1 和 X_2 是统计相关的随机变量，熵为 $H(X_1)=H(X_2)=3$，联合熵为 $H(X_1,X_2)=4<H(X_1)+H(X_2)$。每条边的容量为 $1+\epsilon$，$\epsilon>0$。本例中，为了弄清如何合理地压缩源 X_1 和 X_2，节点 1 和 2 需要知道关于网络拓扑的一些情况。同样地，节点 1 和 2 的输出必须保持统计相关。为了决定如何对这些值进行再编码，节点 3 和 4 需要

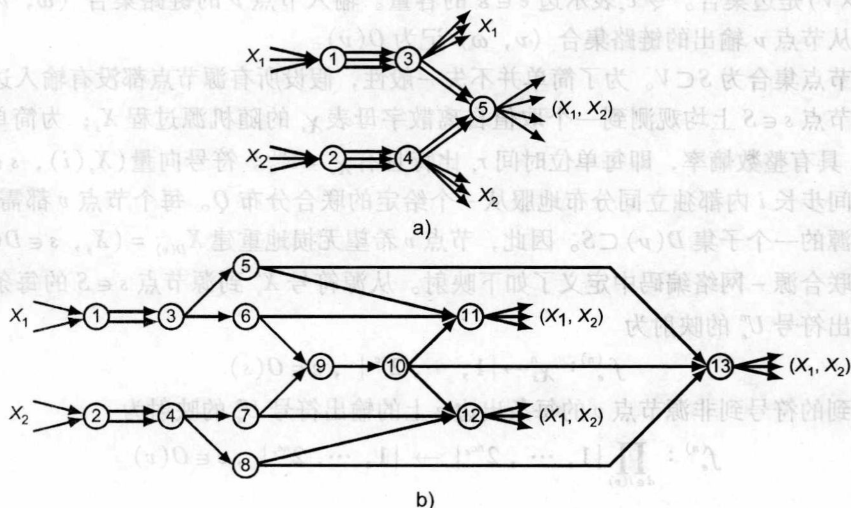


图 8-1 源和网络编码分离失败的示例网络

a）参考文献 [16] 中给出的非组播示例的实例。源 X_1 和 X_2 是统计相关的随机变量，熵为 $H(X_1)=H(X_2)=3$，联合熵为 $H(X_1, X_2)=4$ b）参考文献 [17] 中给出的包含 2 个发送机和 3 个接收机的组播示例。源 X_1 和 X_2 是统计相关的随机变量，熵为 $H(X_1)=H(X_2)=2$，联合熵为 $H(X_1, X_2)=3$。两个示例中的每条边的容量都有 $1+\epsilon$，$\epsilon>0$。

知道这些源的联合统计特性。图8-1b重现了参考文献［17］中给出的包含2个发送机和3个接收机的组播示例。其中，源 X_1 和 X_2 是统计相关的随机变量，熵为 $H(X_1)=H(X_2)=2$，联合熵为 $H(X_1, X_2)=3$，每条边的容量为 $1+\epsilon$。直观地看，蝶形网络从每个数据源发送一个比特给位于节点11和12上的接收机。这些接收机中的每一个都从节点5或8接收到第3个比特，所以节点5和8不能运送相同的信息到蝶形网络中。不幸的是，在分离策略下，节点3由于不能分离在通过蝶形网络的瓶颈时被混合在一起的比特而无法解码。与此相反，采用联合源－网络编码方案能够可靠地将数据源传输给所有预期的接收机，参考文献［15］中首次证明了这一点，详细内容将在8.4节中讨论。

由于即使是一些非常小的网络源编码问题都尚未完全阐明[18]，而且网络编码领域仍充斥着许多NP－困难问题的例子[19-21]，因此我们在联合源网络编码领域的知识仍十分有限也就不足为怪了。本章的其余部分主要关注在一般网络拓扑结构下的相关信源的无损联合源－网络问题。8.2节给出了正式的问题论述并定义了必要的符号。8.3节给出了对网络容量区域的一般概述。8.4节介绍了在一般联合源－网络编码问题的少数特殊情况下的编码策略以及可实现的速率域，我们在这方面的知识相对丰富。

8.2 模型和符号

考虑一个链路无噪的有向无环网络，用图（V，ε）表示，其中 V 是顶点集合，$\varepsilon\subseteq(V\times V)$ 是边集合。令 c_e 表示边 $e\in\varepsilon$ 的容量。输入节点 ν 的链路集合（ω，ν）记为 $I(\nu)$，从节点 ν 输出的链路集合（ν，ω）记为 $O(\nu)$。

源节点集合为 $S\subset V$。为了简单并不失一般性，假设所有源节点都没有输入边㊀。在每个源节点 $s\in S$ 上均观测到一个取值自离散字母表 χ_s 的随机源过程 X_s；为简单起见，假设 X_s 具有整数熵率，即每单位时间 r_s 比特且有 $\chi_s=\mathbb{F}_2^{r_s}$。符号向量 $(X_s(i),\ s\in S)$ 在每个时间步长 i 内都独立同分布地服从一个给定的联合分布 Q。每个节点 v 都需要网络中所有源的一个子集 $D(\nu)\subset S$。因此，节点 v 希望无损地重建 $X_{D(v)}=(X_s,\ s\in D(v))$㊁。

在联合源－网络编码中定义了如下映射。从源符号 X_s^n 到源节点 $s\in S$ 的每条出边 e 上的输出符号 U_e^n 的映射为

$$f_e^{(n)}:\ \chi_s^n\to\{1,\cdots,2^{nc_e}\}\ ,e\in O(s)$$

从接收到的符号到非源节点 v 的每条出边 e 上的输出符号 U_e^n 的映射为

$$f_e^{(n)}:\ \prod_{d\in I(v)}\{1,\cdots,2^{nc_d}\}\to\{1,\cdots,2^{nc_e}\}\ ,e\in O(v)$$

㊀ 如果源节点 ν 存在输入边，则通过创建一个虚拟源节点 ν'，该节点没有输入边并且仅有一条无限容量的输出边（ν'，ν），就可以得到容量域相同的网络。让 ν' 替代 ν 作为源能够获得满足约束且容量域相同的网络。

㊁ 同样，为了简单又不失一般性，每个源节点只能观测到一个单一的源，无论该源是被网络中的其他所有节点需要还是不被任何节点需要。我们可以通过如前所述的方法，对每个不同的信号源创建虚拟节点，来处理单个源节点有多个数据源的情形。

从接收符号到每个汇聚节点 $t\in V$（D（t）$\neq\emptyset$）上的解码符号的解码映射为

$$g_t^{(n)}:\prod_{d\in I(t)}\{1,\cdots,2^{nc_d}\}\rightarrow\prod_{s\in D(t)}\chi_s^n$$

由于网络是无环的，因此可以按照拓扑顺序进行节点映射，从而每个节点都可以在接收到来自其所入边的输入符号后再进行与出边对应的映射。

我们聚焦于无损重建，该重建要求解码错误概率随着编码块长度 n 的无限增大而趋于零。源编码和网络编码相关文献的目标是描述一个给定的源集合何时可以通过一个特定的无噪链路网络进行无损传输，并设计实现在实际限制下有效的通信策略。虽然决定需求可行性的基本目标是相同的，并且这两个领域都通过描述可达速率域来研究该可行性问题，但这两类文献中对“速率域”的定义是不同的。网络拓扑结构对于二者都是固定的。源编码速率域通常先固定源的分布，然后表征了能够同时满足所有需求 $D(t)$，$t\in V$ 的链路容量向量（c_e，$e\in\varepsilon$）。相比之下，网络编码相关文献通常先固定链路容量，然后描述同时满足所有需求 $D(t)$，$t\in V$ 的速率向量（r_s，$s\in S$）。本章采用的是源编码的描述方式，其正式定义见下文。这样的选择是出于如下考虑：对于相关的源，即使是熵的全向量（$H(X_{S'})$，$S'\subseteq S$）也不足以描述传输源（X_s，$s\in S$）所需的链路容量；与此相反，在独立源的情况下，源熵 r_s 已足够表征给定的要求集合的可实现性（参见参考文献［22］中的推论 2.3.3）。为了说明这一情况，考虑图 8-2 所示的例子。源 $X_S=(X_1,X_2)$ 和 $\hat{X}_S=(\hat{X}_1,\hat{X}_2)$ 具有相同的熵向量：

$$(H(X_1),H(X_2),H(X_1,X_2))=(H(\hat{X}_1),H(\hat{X}_2),H(\hat{X}_1,\hat{X}_2))$$

当节点 1 和 2 的输入分别为 X_1 和 X_2 时，节点 3 可以满足要求 $D(3)=\{1\}$。相反，当节点 1 和 2 的输入分别为 $\hat{X}_1$ 和 $\hat{X}_2$ 时，节点 3 无法满足要求 $D(3)=\{1\}$，[2]。

给定一个网络（V，ε），其源 $S\subseteq V$ 服从分布 Q，对于要求 $D(t)\subseteq S$，$\forall t\in V$，我们用 R 来表示可满足给定要求的边容量向量 $\boldsymbol{c}=(c_e,\ e\in\varepsilon)$ 的集合。如果我们限定所有 $e\in\varepsilon$ 的 $f_e^{(n)}$ 都是线性函数，则可以使用 R_L 来表示相应的容量向量的集合。但这不能充分表征所有源和所有要求的速率域 R 和 R_L。8.3 节概括了联合源 - 网络源码的各种已知的性质和界。

8.3　一般联合信源网络编码的速率区域特性

虽然我们对各种特例已有了较好的认识，但对于一般的联合 - 源网络编码的可达速率域却知之甚少。下面简单地概述已有的研究结果。

考虑源为 S 的网络（V，ε），$\forall t\in V$，其要求为 $D(t)$。假设对于每个源 $s\in S$ 和满足 $s\in D(t)$ 的每个汇聚节点 $t\in V$，存在一条从节点 s 到节点 t 的路径。在轻微的失真测度下，无损速率域 R 等于失真测度为 0 时测得的有损速率域[23]（参见参考文献［22］中的定理 2.5.13）。此外，速率域 R 基于基本源分布 Q 连续变化。这个性质非常重要，因为在实际中真正的源分布很少已知。相反，我们在过去的源样本的基础上，对源分布

进行经验估计。源分布的速率域的连续性意味着当对源分布的估计误差逐渐减小时，相应的速率域误差也逐渐减小。

虽然一般情况下的速率域问题尚未得到解决，并且可能很难计算[19, 20]，但是在统计相关源的假设下，有各种工具可用于计算速率域的界。其中最主要的是接下来将描述的割集界。S'和t之间的一个割将V分成了两个集合A和$V\setminus A$，满足$S'\subset A$以及$t\in V\setminus A$。这个割的容量等于那些起始节点属于A、结束节点属于$V\setminus A$的边的容量之和。对于给定的边容量向量$\boldsymbol{c}$，我们用$m_{S',t}(\boldsymbol{c})$表示源的一个子集$S'\subseteq S$与汇聚节点t之间的最小割容量。对于源集合$S'\subseteq D(t)$和汇聚节点t之间的任意割A，描述从S'中的源到汇聚节点t所需的最小速率为$H(X_{S'}\mid X_{S\setminus S'})$，该速率也是根据斯莱皮恩－沃尔夫定理(Sleipian－Wolf)[24]得到的，它描述从节点$X_S{}'$到一个已知$X_{S\setminus S'}$的节点所需的速率（参见参考文献［25］中的15.10节）。令R_C表示满足割集界的边容量向量的集合：

$$m_{S',t}(\boldsymbol{c})>H(X_{S'}\mid X_{S\setminus S'})\ \forall S'\subset D(t),t\in V \tag{8-1}$$

对于一个给定的网络（V，$\boldsymbol{\varepsilon}$）和源分布Q，其割集界为无损源编码问题提供了一个速率域的外界。确切地说，对于任意一个固定的源节点为S且需求为$D(t)$的网络（V，$\boldsymbol{\varepsilon}$），并对于任意固定的源分布Q，有

$$R_L\subseteq R\subseteq R_C$$

一般来说，割集界不是紧界。例如，图8-2所示的编码边信息问题就是一个非常小的网络的例子，对于源（X_1，X_2）上的所有可能的分布，该网络的割集界无法均等地实现。8.4节将描述一系列其割集界是紧界的联合源－网络编码问题，同时也将讨论在实践中能够接近该最优极限性能的各种机制。

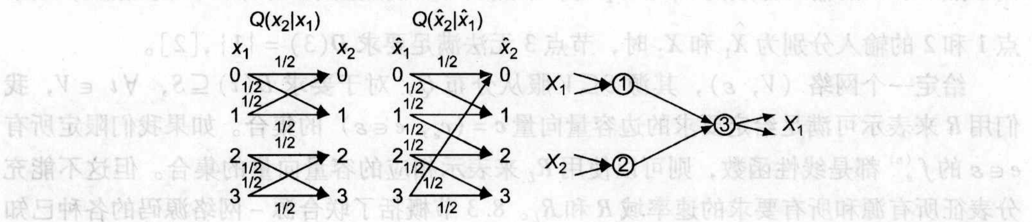


图8-2　即使熵向量（$H(X_1)$，$H(X_2)$，$H(X_1, X_2)$）和（$H(\hat{X}_1)$，$H(\hat{X}_2)$，$H(\hat{X}_1, \hat{X}_2)$）相同，源（X_1，X_2）和（$\hat{X}_1$，$\hat{X}_2$）的速率域也可能不同，如上述示例所示。源X_1和$\hat{X}_1$独立同分布地取自字符集$\{0, 1, 2, 3\}$，源X_2和$\hat{X}_2$分如遵循如上定义的分布$Q(x_2\mid x_1)$和$Q(\hat{x}_2\mid \hat{x}_1)$。这两类源的码边信息问题的速率域不同。例如，当$p=0.5$且每条链路的容量为$1+\epsilon$时，源（$X_1$，$X_2$）可以满足给定的需求，但源（$\hat{X}_1$，$\hat{X}_2$）不能满足。

8.4　无损组播的容量结果

本节中我们将描述当统计相关源下的联合源－网络编码的容量域已知时的两种设定。这两个例子都属于多源组播问题，其中存在一些源的集合$A\subseteq S$和一些汇聚节点的集合T满足：$\forall t\in T$都有$D(t)=A$，以及$\forall v\notin T$都有$D(v)=\varnothing$。在第一种情况下，$A=$

S；而在第二种情况下，A 是 S 的一个严格子集，其余的源都可以看做是接收机上的边信息。

8.4.1　无边信息场景

我们首先考虑没有边信息的场景，即网络中所有的信息均源自源节点并被所有的汇聚节点需求。在这种情形下，容量域由割集特征所决定，并可通过线性网络编码和非线性解码加以实现。

该结论的正式表述如下。

定理 8-1　$R = R_L = R_C$。

由式（8-1）所定义的区域 R_C 与每个单独的汇聚节点的 Slepian - Wolf 容量域相对应。上述定理表明，利用线性网络编码时，任意速率矢量，只要能被每个单独的汇聚节点实现，它就可以被所有的汇聚节点同时实现。请注意，不同的汇聚节点可以工作在 Slepian - Wolf 域的不同的点上，因此统计相关的源可以相互作为对方的部分"镜像站点"。

为了展示使用线性编码时域 R_C 的可行性，我们考虑对由来自每个源 s 的 nr_s 个比特组成的块进行随机向量线性编码。具体而言，源节点 s 的出边 e 上的符号由下式给出：

$$U_e^n = \boldsymbol{F}_{s,e} X_s^n,\ e \in O(s)$$

非源节点 v 的出边 e 上的符号由下式给出：

$$U_e^n = \sum_{d \in T(v)} \boldsymbol{F}_{d,e} U_d^n,\ e \in O(v)$$

其中，$\boldsymbol{F}_{s,e}$ 和 $\boldsymbol{F}_{d,e}$ 分别是维度为 $nr_s \times nc_e$ 和 $nc_d \times nc_e$ 的二进制矩阵，其元素均独立均匀地随机选取自 $\mathbb{F}_2$。解码函数是非线性的。每个汇聚节点都将它接收到的符号映射为一个解码符号向量，该向量在所有可能的与接收到的符号对应的源符号矢量中具有最小熵或最大的 Q - 概率。每个汇聚节点的解码错误概率的上界都由割容量 $m_{S',t}(c)$ 和源 - 接收机路径的最大长度 L 所决定，该上界随块长度 n 的增大渐近为零。为简便起见，这里省略了证明过程，感兴趣的读者可以参考附录。

8.4.2　汇聚节点具有边信息的场景

对容量已知的上述场景的一般化允许汇聚节点掌握边信息。该模型中我们假设每个汇聚节点 $t \in T$ 都没有出边并可以观察到一个取值自离散值字母表 Y_t 的边信息随机过程 Y_t，其他都与 8.2 节中描述的相同。汇聚节点没有出边的假设避免了更为复杂的需要考虑编码和发送边信息的情况。每个时间步长 i 内的符号矢量（$X_s(i)$：$s \in S, Y_t(i)$：$t \in T$）都独立同分布地遵循一个给定的联合分布 Q。每个汇聚节点都要求所有的源信息而不是边信息。R_C 的定义被相应地修改为满足割集界的速率向量的集合：

$$m_{S',t}(\boldsymbol{c}) > H(X_{S'} \mid X_{S \setminus S'},\ Y_t)\ \forall S' \subseteq S,\ t \in T$$

定理 8-2　$R = R_L = R_C$。

证明：由于每个可实现的速率矢量 $\boldsymbol{c}$ 都满足割集界，因此有 $R \subseteq R_C$，从而可得 $R_L \subseteq$

$R \subseteq R_C$。

要证明等号成立，只要证明 $R_L = R_C$ 即可。我们证明了要实现每个汇聚节点上的错误概率的消失，其一系列随机线性码需满足的充分条件为 $\boldsymbol{c} \in R_C$。

对于 $t \in T$，令（V_t，ε_t）表示删去除 t 之外的其他所有汇聚节点得到的网络。因此有，$V_t = (V \backslash T) \cup \{t\}$ 以及 $\varepsilon_t = \varepsilon \backslash Ut' \in J \backslash \{t\} I(t')$。注意，由于 Y_t 对于汇聚节点 t 是已知的，因此将网络（V_t，ε_t）看做是一个需求为（X_S，Y_t）的组播网络无损于最优性。

对于速率矢量 $\boldsymbol{c}$，令 $\boldsymbol{c}_t$ 表示 $\boldsymbol{c}$ 对网络（V_t，ε_t）的约束。由于从节点 $T \backslash \{t\}$ 到节点 t 之间不存在直达路径，因此删除集合 $T \backslash \{t\}$ 中的节点不会影响从 S 的任意子集 S' 到 t 的最大流，从而其最小割不变，即

$$m_{S',t}(\boldsymbol{c}) = m_{S',t}(\boldsymbol{c}_t) \; \forall S' \subseteq S$$

在采用定理 8-1 中的随机线性码构造方法构造的网络（V，ε）中，对于块长度 n，令$(f_e^{(n)}:\ e \in \varepsilon)$和$(g_t^{(n)}:\ t \in T)$分别表示一个速率为 $\boldsymbol{c}$ 的随机码的编码映射和解码映射。由于编码构造只依赖于边容量，因此映射$(f_e^{(n)}:\ e \in \varepsilon_t)$和 $g_t^{(n)}$ 还定义了网络（V_t，ε_t）中速率为 $\boldsymbol{c}_t$ 的随机线性码。因此由定理 8-1 可知，随着块长度 n 无界地增大，（V_t，ε_t）中的错误概率消失的充分条件为

$$m_{S',t}(\boldsymbol{c}_t) > H(X_{S'} \mid X_{S \backslash S'}, Y_t) \; \forall S' \subseteq S$$

最后我们注意到：

$$P_r(g_t^{(n)}(X_S^n, Y_t^n) \neq X_S^n, \exists t \in T) \leqslant \sum_{t \in T} P_r(g_t^{(n)}(X_S^n, Y_t^n) \neq X_S^n)$$

因此，随着块长度 n 无界地增大，全部错误概率消失的充分条件为

$$m_{S',t}(\boldsymbol{c}_t) > H(X_{S'} \mid X_{S \backslash S'}, Y_t) \; \forall S' \subseteq S, \forall t \in T$$

由于 $m_{S',t}(\boldsymbol{c}_t) = m_{S',t}(\boldsymbol{c})$，这表明通过线性编码每个速率矢量 $\boldsymbol{c} \in R_C$ 都可以实现，故有 $R_C \subseteq R_L$。证毕。

8.5 实用的方法

已经有一些旨在结合网络编码和压缩工作，其中一些采用了一种模拟方法来编码。参考文献［26］和［27］中详细地说明了这种方法，它在本质上是一种基于失真的方法，与参考文献［28］和［29］中在进行视频分发时（甚至不使用网络编码）采用的方法相似，它使用简单的源/网络编码来实现 8.3 节中给出的原理。问题在于码字在模拟域上被组合时，节点的失真也会被累加。

在无损组播领域，我们面临的主要的挑战之一是提出一种无需传统解码的高复杂度和开销的联合源－网络编码。已经提出的一个方案是在 Turbo 码的基础上建立能够结合已知的解码技术[30]的良好性能的源－网络码，即使这些技术的复杂度不能被简单地表征。另一个挑战是确定对于不同的源我们应以什么程度削减冗余。冗余的过度削减将不允许系统利用源数据的冗余所提供的虚拟镜像系统的优势，若冗余削减不足，又将导致发送无用的自由度。参考文献［31］和［32］分别考虑了源间的冗余是自然地产生和设

计产生的情况。后者表明了使用网络编码来生成镜像站点的可能性，迄今为止镜像站点尚依赖于使用复制的文件。

8.6 附录

下面讨论随机向量线性网络编码的错误概率。为简单起见，我们将阐述并证明在 2 个源的情形下的错误概率界。

证明过程使用了类型方法，是对用于分析线性 Slepian - Wolf 编码的 Csiszár 方法[33]的一般化。矢量 $\boldsymbol{x}\in\mathbb{F}_2^n$ 的类型 $P_{\mathbf{x}}$ 分布在$\mathbb{F}_2$上，由$\mathbb{F}_2$上的 $\boldsymbol{x}$ 中的元素的相对频率来定义。对联合类型 $P_{\mathbf{xy}}$的定义与之类似。

定理 8-3 每个汇聚节点上的解码错误概率均不大于$\sum_{i=1}^{3}p_e^i$，其中：

$$p_e^1\leqslant\exp\left\{-n\min_{X,Y}\left(D(P_{XY}\parallel Q)+|m_1\left(1-\frac{1}{n}\log L\right)-H(X|Y)|^+\right)+2^{2r_1+r_2}\log(n+1)\right\}$$

$$p_e^2\leqslant\exp\left\{-n\min_{X,Y}\left(D(P_{XY}\parallel Q)+|m_2\left(1-\frac{1}{n}\log L\right)-H(Y|X)|^+\right)+2^{r_1+2r_2}\log(n+1)\right\}$$

$$p_e^3\leqslant\exp\left\{-n\min_{X,Y}\left(D(P_{XY}\parallel Q)+|m_3\left(1-\frac{1}{n}\log L\right)-H(XY)|^+\right)+2^{2r_1+2r_2}\log(n+1)\right\}$$

且 X 和 Y 是联合分布为 P_{XY}的伪随机变量。

证明：这里仅给出大体的证明，证明的全过程可参看参考文献［15］。

考虑源矢量［$\boldsymbol{x}_1$，$\boldsymbol{x}_2$］$\in\mathbb{F}_2^{n(r_1+r_2)}$ 的传输。传输矩阵 $\boldsymbol{C}_{I(t)}$ 给出了从源向量［$\boldsymbol{x}_1$，$\boldsymbol{x}_2$］到汇入汇聚节点 t 的终点集合 $I(t)$ 上的比特矢量 $\boldsymbol{z}$ 的映射。

节点 t 上的解码函数将接收到的比特矢量 $\boldsymbol{z}$ 映射为可使 $\alpha(P_{\mathbf{x}_1\mathbf{x}_2})$ 最小化且满足［$\boldsymbol{x}_1$，$\boldsymbol{x}_2$］$\boldsymbol{C}_{I(t)}=\boldsymbol{z}$ 的矢量［$\tilde{\boldsymbol{x}}_1$，$\tilde{\boldsymbol{x}}_2$］$\in\mathbb{F}_2^{n(r_1+r_2)}$。对于最小熵解码器有 $\alpha(P_{\mathbf{x}_1\mathbf{x}_2})\equiv H(P_{\mathbf{x}_1\mathbf{x}_2})$，然而对于最大 Q－概率解码器则有 $\alpha(P_{\mathbf{x}_1\mathbf{x}_2})\equiv-\log Q^n(\boldsymbol{x}_1\boldsymbol{x}_2)$。有 3 种类型的解码错误，第 1 种：解码器输出的 $\boldsymbol{x}_2$ 值正确，但 $\boldsymbol{x}_1$ 值错误；第 2 种：解码器输出的 $\boldsymbol{x}_1$ 值正确，但 $\boldsymbol{x}_2$ 值错误；第 3 种：解码器输出的 $\boldsymbol{x}_1$ 和 $\boldsymbol{x}_2$ 的值均错误。我们分别用 p_e^1、p_e^2、p_e^3 表示求得的这 3 种类型错误概率的上界。

序列的（联合）类型可以看做是伪变量 X、Y 等的（联合）分布 P_X($P_{X,Y}$等)。域 $\mathbb{F}_2^k$ 上的序列的不同类型的集合可以用$\mathscr{P}(\mathbb{F}_2^k)$ 表示，我们定义该类型集合如下：

$$\mathscr{P}_n^i=\begin{cases}\{P_{X\tilde{X}Y\tilde{Y}}\in\mathscr{P}(\mathbb{F}_2^{nr_1}\times\mathbb{F}_2^{nr_1}\times\mathbb{F}_2^{nr_2}\times\mathbb{F}_2^{nr_2})\ |\tilde{X}\neq X,\ \tilde{Y}=Y\}\ i=1\\\{P_{X\tilde{X}Y\tilde{Y}}\in\mathscr{P}(\mathbb{F}_2^{nr_1}\times\mathbb{F}_2^{nr_1}\times\mathbb{F}_2^{nr_2}\times\mathbb{F}_2^{nr_2})\ |\tilde{X}\neq X,\ \tilde{Y}=Y\}\ i=2\\\{P_{X\tilde{X}Y\tilde{Y}}\in\mathscr{P}(\mathbb{F}_2^{nr_1}\times\mathbb{F}_2^{nr_1}\times\mathbb{F}_2^{nr_2}\times\mathbb{F}_2^{nr_2})\ |\tilde{X}\neq X,\ \tilde{Y}=Y\}\ i=3\end{cases}$$

其中在上述定义中的 $\tilde{X}\neq X(\tilde{Y}=Y)$ 表明定义类型的基本向量各不相同。

我们还定义了序列集合：

$$\mathscr{T}_{XY}=\left\{[\boldsymbol{x}_1,\boldsymbol{x}_2]\in\mathbb{F}_2^{n(r_1+r_2)}\mid P_{\mathbf{x}_1\mathbf{x}_2}=P_{XY}\right\}$$

$$\mathscr{T}_{\tilde{X}\tilde{Y}|XY}(\boldsymbol{x}_1\boldsymbol{x}_2)=\left\{[\tilde{\boldsymbol{x}}_1,\tilde{\boldsymbol{x}}_2]\in\mathbb{F}_2^{n(r_1+r_2)}\mid P_{\mathbf{x}_1\tilde{\mathbf{x}}_2\mathbf{x}_1\mathbf{x}_2}=P_{\tilde{X}\tilde{Y}XY}\right\}$$

接着，我们可以将错误概率的表达式写为

$$p_e^1\leqslant\sum_{\substack{P_{X\tilde{X}Y\tilde{Y}}\in\mathscr{P}_n^1:\\ \alpha(P_{\tilde{X}Y})\leqslant\alpha(P_{XY})}}\sum_{(\mathbf{x}_1,\mathbf{x}_2)\in\mathscr{T}_{XY}}Q^n(\boldsymbol{x}_1\boldsymbol{x}_2)\times Pr(\exists(\tilde{\boldsymbol{x}}_1,\tilde{\boldsymbol{x}}_2)\in\mathscr{T}_{\tilde{X}\tilde{Y}|XY}(\boldsymbol{x}_1\boldsymbol{x}_2)\text{ s. t. }[\boldsymbol{x}_1-\tilde{\boldsymbol{x}}_1,0]C_{I(t)}=0)\leqslant$$

$$\sum_{\substack{P_{X\tilde{X}Y\tilde{Y}}\in\mathscr{P}_n^1:\\ \alpha(P_{\tilde{X}Y})\leqslant\alpha(P_{XY})}}\sum_{(\mathbf{x}_1,\mathbf{x}_2)\in\mathscr{T}_{XY}}Q^n(\boldsymbol{x}_1\boldsymbol{x}_2)\times\min\left\{\sum_{(\mathbf{x}_1,\mathbf{x}_2)\in\mathscr{T}_{\tilde{X}\tilde{Y}|XY}(\mathbf{x}_1\mathbf{x}_2)}Pr([\boldsymbol{x}_1-\tilde{\boldsymbol{x}}_1,0]C_{I(t)}=0),1\right\}$$

$$p_e^2\leqslant\sum_{\substack{P_{X\tilde{X}Y\tilde{Y}}\in\mathscr{P}_n^2:\\ \alpha(P_{\tilde{X}Y})\leqslant\alpha(P_{XY})}}\sum_{(\mathbf{x}_1,\mathbf{x}_2)\in\mathscr{T}_{XY}}Q^n(\boldsymbol{x}_1\boldsymbol{x}_2)\times\min\left\{\sum_{(\mathbf{x}_1,\mathbf{x}_2)\in\mathscr{T}_{\tilde{X}\tilde{Y}|XY}(\mathbf{x}_1\mathbf{x}_2)}Pr([0,\boldsymbol{x}_2-\tilde{\boldsymbol{x}}_2]C_{I(t)}=0),1\right\}$$

$$p_e^3\leqslant\sum_{\substack{P_{X\tilde{X}Y\tilde{Y}}\in\mathscr{P}_n^3:\\ \alpha(P_{\tilde{X}\tilde{Y}})\leqslant\alpha(P_{XY})}}\sum_{(\mathbf{x}_1,\mathbf{x}_2)\in\mathscr{T}_{XY}}Q^n(\boldsymbol{x}_1\boldsymbol{x}_2)\times\min\left\{\sum_{(\mathbf{x}_1,\mathbf{x}_2)\in\mathscr{T}_{\tilde{X}\tilde{Y}|XY}(\mathbf{x}_1\mathbf{x}_2)}Pr([\boldsymbol{x}_1-\tilde{\boldsymbol{x}}_1,\boldsymbol{x}_2-\tilde{\boldsymbol{x}}_2]C_{I(t)}=0),1\right\}$$

其中的概率取自与随机网络码相应的网络传输矩阵 $\boldsymbol{C}_{I(t)}$ 的实现。

我们可以通过使用 $|\mathscr{P}_n^i|$ 、$|\mathscr{T}_{XY}|$ 和 $\mathscr{T}_{\tilde{X}\tilde{Y}|XY}(\boldsymbol{x}_1\boldsymbol{x}_2)$ 的简单且相同的基数界：

$$Q^n(\boldsymbol{x}_1\boldsymbol{x}_2)=\exp\{-n(D(P_{XY}\parallel Q)+H(XY))\}$$

$$(\boldsymbol{x}_1,\boldsymbol{x}_2)\in\mathscr{T}_{XY}$$

得到：

$$p_e^1\leqslant\exp\left\{-n\min_{\substack{P_{X\tilde{X}Y\tilde{Y}}\in\mathscr{P}_n^1:\\ \alpha(P_{\tilde{X}Y})\leqslant\alpha(P_{XY})}}\left(D(P_{XY}\parallel Q)+\left|-\frac{1}{n}\log P_1-H(\tilde{X}|XY)\right|^+\right)+2^{2r_1+r_2}\log(n+1)\right\}$$

$$p_e^2\leqslant\exp\left\{-n\min_{\substack{P_{X\tilde{X}Y\tilde{Y}}\in\mathscr{P}_n^2:\\ \alpha(P_{X\tilde{Y}})\leqslant\alpha(P_{XY})}}\left(D(P_{XY}\parallel Q)+\left|-\frac{1}{n}\log P_2-H(\tilde{Y}|XY)\right|^+\right)+2^{r_1+2r_2}\log(n+1)\right\}$$

$$p_e^3\leqslant\exp\left\{-n\min_{\substack{P_{X\tilde{X}Y\tilde{Y}}\in\mathscr{P}_n^3:\\ \alpha(P_{\tilde{X}\tilde{Y}})\leqslant\alpha(P_{XY})}}\left(D(P_{XY}\parallel Q)+\left|-\frac{1}{n}\log P_3-H(\tilde{X}\tilde{Y}|XY)\right|^+\right)+2^{2r_1+2r_2}\log(n+1)\right\}$$

其中的指数和对数的底均为2，且有：

$$P_1 = P_r([\boldsymbol{x}_1 - \tilde{\boldsymbol{x}}_1, 0]\boldsymbol{C}_{I(t)} = 0)$$

$$P_2 = P_r([0, \boldsymbol{x}_2 - \tilde{\boldsymbol{x}}_2]\boldsymbol{C}_{I(t)} = 0)$$

$P_3 = P_r([\boldsymbol{x}_1 - \tilde{\boldsymbol{x}}_1, \boldsymbol{x}_2 - \tilde{\boldsymbol{x}}_2]\boldsymbol{C}_{I(t)} = 0), \boldsymbol{x}_1 - \tilde{\boldsymbol{x}}_1, \boldsymbol{x}_2 - \tilde{\boldsymbol{x}}_2 \neq 0$。

P_1、P_2和 P_3表示两个不同的源序列（一个或两个源节点中的值都不相同）在汇聚节点由网络编码映射到相同的观测序列的概率；请注意，使用随机线性网络编码时，对于任何非零的 $\boldsymbol{x}_1 - \tilde{\boldsymbol{x}}_1$，$\boldsymbol{x}_2 - \tilde{\boldsymbol{x}}_2$，该概率固定不变。这些概率的界与如下参数有关：块长度 n、网络参数 m_i（其中 $i = 1, 2$ 时，m_i 表示接收机和源 X_i 之间的最小割容量，m_3 为接收机和两个数据源之间的最小割容量 m_3）以及源与接收机之间的最大路径长度 L，具体地可以写为

$$P_i \leqslant \left(1 - \left(1 - \frac{1}{2^n}\right)^L\right)^{m_i} \leqslant \left(\frac{L}{2^n}\right)^{m_i} \tag{8-2}$$

对于最小熵解码器，我们有：

$$\alpha(P_{\tilde{X}\tilde{Y}}) \leqslant \alpha(P_{XY}) \Rightarrow \begin{cases} H(\tilde{X}|XY) \leqslant H(\tilde{X}|Y) \leqslant H(X|Y) & \text{对于 } Y = \tilde{Y} \\ H(\tilde{Y}|XY) \leqslant H(\tilde{Y}|X) \leqslant H(Y|X) & \text{对于 } X = \tilde{X} \\ H(\tilde{X}\tilde{Y}|XY) \leqslant H(\tilde{X}\tilde{Y}) \leqslant H(XY) \end{cases}$$

则可得：

$$p_e^1 \leqslant \exp\left\{-n \min_{XY}\left(D(P_{XY} \| Q) + \left| -\frac{1}{n}\log P_1 - H(X|Y)\right|^+\right) + 2^{2r_1 + r_2}\log(n+1)\right\}$$

$$p_e^2 \leqslant \exp\left\{-n \min_{XY}\left(D(P_{XY} \| Q) + \left| -\frac{1}{n}\log P_2 - H(Y|X)\right|^+\right) + 2^{r_1 + 2r_2}\log(n+1)\right\}$$

$$p_e^3 \leqslant \exp\left\{-n \min_{XY}\left(D(P_{XY} \| Q) + \left| -\frac{1}{n}\log P_3 - H(XY)\right|^+\right) + 2^{2r_1 + 2r_2}\log(n+1)\right\}$$

以上还表明对于最大 Q - 概率解码器也有同样的界。将式（8-2）中的 P_i的界代入上式中即可得到所需结果。

证毕。

参考文献

[1] D. Slepian and J. K. Wolf. Noiseless coding of correlated information sources. IEEE Transactions on Information Theory, 19: 471 – 480, 1973.

[2] R. Ahlswede and J. Ko¨ rner. Source coding with side information and a converse for degraded broadcast channels. IEEE Transactions on Information Theory, 21: 629 – 637, Nov. 1975.

[3] H. Yamamoto. Wyner – Ziv theory for a general function of the correlated sources. IEEE Transactions on Information Theory, 28: 1788 – 1791, Sept. 1982.

[4] A. Orlitsky and J. R. Roche. Coding for computing. IEEE Transactions on Information Theory, 47: 903 – 917, Mar. 2001.

[5] H. S. Witsenhausen. The zero – error side information problem and chromatic numbers. IEEE Transactions on Information Theory, 22: 592 – 593, 1976.

[6] N. Alon and A. Orlitsky. Source coding and graph entropies. IEEE Transactions on Information Theory, 42: 1329 – 1339, Sept. 1996.

[7] A. D. Wyner and J. Ziv. The rate – distortion function for source coding with side infor – mation at the decoder. IEEE Transactions on Information Theory, 22: 1 – 10, Jan. 1976.

[8] T. Berger and S. Y. Tung. Encoding of correlated analog sources. In Proceedings of IEEE – USSR Joint Workshop on Information Theory, pp. 7 – 10, 1975.

[9] R. M. Gray and A. D. Wyner. Source coding for a simple network. Bell Systems Technical Journal, 53: 1681 – 1721, Nov. 1974.

[10] A. D. Wyner. On source coding with side information at the decoder. IEEE Transactions on Information Theory, 21: 294 – 300, Nov. 1975.

[11] T. Berger and R. Yeung. Multiterminal source encoding with one distortion criterion. IEEE Transactions on Information Theory, 35 (2): 228 – 236, 1989.

[12] R. Ahlswede, N. Cai, S. – Y. R. Li, and R. W. Yeung. Network information flow. IEEE Transactions on Information Theory, 46: 1204 – 1216, July 2000.

[13] R. Koetter and M. Me′dard. An algebraic approach to network coding. IEEE/ACM Transactions on Networking, 11: 782 – 795, Oct. 2003.

[14] L. Song, R. W. Yeung, and N. Cai. Zero – error network coding for acyclic networks. IEEE Transactions on Information Theory, 49: 3129 – 3139, July 2003.

[15] T. Ho, M. M′edard, R. Koetter, D. R. Karger, M. Effros, J. Shi, and B. Leong. A random linear network coding approach to multicast. IEEE Transactions on Information Theory, 52: 4413 – 4430, Oct. 2006.

[16] M. Effros, M. M′edard, T. Ho, S. Ray, D. Karger, and R. Koetter. Linear network codes: A unified framework for source, channel, and network coding. In Proceedings of the DIMACS Workshop on NetworkInformation Theory, (Piscataway, NJ), IEEE, Mar. 2003. Invited paper.

[17] A. Ramamoorthy, K. Jain, P. A. Chou, and M. Effros. Separating distributed source coding from network coding. Joint Special Issue of the IEEE Transactions on Information Theory and the IEEE/ACM Transactions on Networking, 52: 2785 – 2795, June 2006.

[18] T. Berger. Multiterminal source coding. In The Information Theory Approach to Communications, (New York), pp. 172 – 231, CISM Courses and Lectures, Springer – Verlag, July 1977.

[19] A. R. Lehman and E. Lehman. Complexity classifications of network information flow problems. In Allerton Annual Conference on Communications, Control, and Computing, (Monticello, IL), Sept. 2003.

[20] M. Langberg, M. Sprintson, and J. Bruck. Network coding: A computational perspective. IEEE Transactions on Information Theory, 55 (1): 145 – 157, 2008.

[21] H. Yao and E. Verbin. Network coding is highly non – approximable. In Allerton Annual Conference on Communications, Control, and Computing, 2009.

[22] W. – H. Gu. On achievable rate regions for source coding networks. Ph. D. dissertation, California

Institute of Technology, Pasadena, CA, 2009.

[23] W. – H. Gu and M. Effros. On the concavity of rate regions for lossless source coding in networks. In Proceedings of the IEEE International Symposium on Information Theory, (Seattle, WA), pp. 1599 – 1603, July 2006.

[24] D. Slepian and J. K. Wolf. A coding theorem for multiple access channels with correlated information sources. Bell System Technical Journal, 52: 1037 – 1076, 1973.

[25] T. M. Cover and J. A. Thomas. Elements of Information Theory. Wiley, second ed. , 2006.

[26] S. Katti, S. Shintre, S. Jaggi, D. Katabi, and M. M'edard. Real network codes breaking the all – or – nothing barrier. In Allerton Annual Conference on Communications, Control, and Computing, 2007.

[27] B. Dey, S. Katti, S. Jaggi, D. Katabi, and M. M'edard. "Real" and "complex" network codes—promises and challenges. In Fourth Workshop on Network Coding Theory and Applications (NETCOD), 2008.

[28] G. Woo, D. Katabi, and S. Chachulski. One video stream to serve diverse receivers. Tech. rep. , MIT, 2008.

[29] D. Katabi, R. Hariharan, and S. Jakubczak. Softcast: One video to serve all wireless receivers. Mit – csail – tr – 2009 – 005, MIT, 2009.

[30] G. Maierbacher, J. Barros, and M. M'edard. Practical sourcenetwork decoding. In IEEE International Symposium on Wireless Communication Systems, 09.

[31] A. Lee, M. M'edard, K. Z. Haigh, S. Gowan, and P. Rubel. Minimumcost subgraphs for joint distributed source and network coding. In Third Workshop on Network Coding, Theory, and Applications, 2007.

[32] S. Huang, A. Ramamoorthy, and M. M'edard. Minimum cost mirror sites using network coding: Replication vs. coding at the source nodes. IEEE Transactions on Information Theory, 57 (2): 2011.

[33] I. Csisza'r. Linear codes for sources and source networks: Error exponents. IEEE Transactions on Information Theory, 28: 585 – 592, July 1982.

第 9 章　网络编码的标度律

Atilla Eryilmaz 美国，俄亥俄州，哥伦布，俄亥俄州立大学，电子与计算机工程系
Lei Ying 美国，爱荷华州，艾姆斯，爱荷华州立大学，电子与计算机工程系

摘要

本章概述了当放大或缩小网络规模或编码窗口大小时与网络编码相关的标度律结果。这些结果突出显示了网络编码在我们感兴趣的各种场景下带来的潜在增益。我们讨论了在信道时变以及节点可移动这两个条件下，吞吐量和延迟这两个无线网络性能指标的标度律。

关键词

标度律、随机网络编码、吞吐量-延迟权衡、移动模型、移动 ad hoc 网络、无线调度。

9.1　引言及基本设定

依据 Claude E. Shannon 在参考文献［27］中建立的框架，长期以来人们都认为编码是用于给通信链路提供吞吐量增益的。编码可以在长块传输信号之间散布信息，从而获得对抗信道统计特性变化的鲁棒性，这也是编码的根本力量。虽然上述思想使得全球通信网络成为可能并改变了世界，但其最激动人心的进展也仅限于点到点链路以及其他一些有限的多播场景下的容量结果。近十年前，网络编码的思想[2,17,19]出现了，它提供了一种在一般网络中通过简单的代数编码策略来获得显著的吞吐量增益的手段。本章中我们主要关注当通信系统采用网络编码策略时，与吞吐量和延迟性能相关的标度律。这些标度律本身以及它们与非编码技术的比较可以帮助我们量化在大规模系统中通过编码获得的增益。

下面我们主要考虑分组通信网络，网络中的每个数据包都是有限域$\mathbb{F}_q$上的一个维度为 m 的向量，用 $\boldsymbol{P}$ 表示。这种结构的代数性质使得我们可以在不同的数据包（即向量）之间进行代数操作，从而获得一个新的该向量空间中的数据包。这种能力使得我们能够在网络中进行（线性）网络编码。在大多数讨论中，我们都假设了时隙化的系统操作：即所有的节点都与一个共同的时钟同步并在宽度固定的时隙内操作。下面我们将依据上述设定来严格地定义网络编码操作。

定义 9-1（(线性)网络编码）（线性)网络编码指的是任意网络节点对可用数据包集合 $\{\boldsymbol{P}_k\}_{k\geq 1}$ 中的元素进行任意线性组合的一种传输模式。具体地说，在第 t 个时隙有：

$$\boldsymbol{P}[t] = \sum_{k\geq 1} a_k[t]\, \boldsymbol{P}_k \tag{9-1}$$

其中，$a_k[t]$ 是全局编码系数，$a_k[t] \in \mathbb{F}_q, \forall k$。

作为比较，我们还定义了另一种极端的运算模式，它只允许节点发送可用数据包集合中的一个元素。这种模式被称为调度模式，将在下文给出详细描述。

定义 9-2（调度）　调度指的是不进行网络编码的传统传输模式：发送者必须从可用数据包集合 $\{\boldsymbol{P}_k\}_{k\geq 1}$ 中选出一个数据包来传输。具体地说，在第 t 个时隙有 $\boldsymbol{P}[t]\in\{\boldsymbol{P}_k: k=1,2,\cdots\}$。

下面首先考虑发送者通过不可靠信道将数据广播给所有的（或一个子集的）接收者的场景，该场景可以看做是蜂窝网络或多跳无线网络中的多播通信的基本构件。我们可以使用消失信道来建模蜂窝或卫星通信中的物理信道，或者建模用端到端传输表示多次级联传输的虚拟信道，例如在多跳无线网络中。我们的讨论分为两种场景，首先将在 9.2 节中学习与静态网络相关的标度律，接下来将在 9.3 节中讨论移动 Ad Hoc 网络的情况。

9.2　有损链路下的无线广播

本节重点关注与有损链路下的一系列无线广播场景相关的吞吐量和延迟的标度律的紧界，分 3 个部分对这个问题进行讨论：（1）延迟标度：讨论了网络编码在延迟方面带来的性能提升并给出了主要的研究结果；（2）扩展：在第（1）部分的基础上进一步考虑了对于网络拓扑、数据流的动态到达以及应用程序的延迟灵敏度的扩展；（3）吞吐量 - 延迟权衡：揭示了早期的编码结果在吞吐量方面的缺陷，并给出了使用网络编码实现的延迟 - 吞吐量标度律的最新结果。

9.2.1　延迟标度增益

图 9-1 所示的是在参考文献［10，11］中被引入并分析的一个下载场景，该场景中需要通过消失信道将 K 个数据包广播给全部 N 个接收者。为了表征信道的变化，我们用 $\{C_n[t]\}_{t\geq 0}$ 表示第 n 个用户的信道状态过程，它是一个期望为 $c_n\in[0,1]$ 的伯努利（Bernoulli）过程。假设各用户的信道状态过程相互独立，当且仅当 $C_n[t]=1$ 时，第 n 个接收者才能成功接收在时隙 t 发送的数据包 $\boldsymbol{P}[t]$。

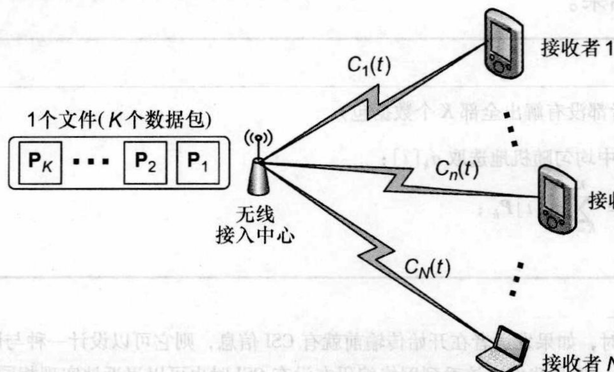


图 9-1　N 个相互独立的消失信道下的无速率广播，其中 C_n［t］服从期望为 C_n 的伯努利分布

对于上述场景，我们需要回答的第一个问题是网络编码是否优于最优调度策略，甚至是在发送者在发送之前就具有完全信道状态信息（CSI）的条件下。这里的 CSI 指的是发送者在任意时隙 t 开始时对实际的 $\boldsymbol{C}[t]\triangleq(C_n[t])_{n=1}^{N}$ 的掌握。下面的例子引自参考文献［10］，它证明了只要 $K\geqslant 3$ 且 $N\geqslant 3$，我们就有可能获得上述增益[⊖]。

例 9-1　考虑 $K=3$ 且 $N=3$ 的情况，即发送者需要把 3 个数据包广播给 3 个接收者。假设信道实现为：$\boldsymbol{C}[1]=(0,1,1)$、$\boldsymbol{C}[2]=(1,0,1)$、$\boldsymbol{C}[3]=(1,1,0)$、$\boldsymbol{C}[4]=(1,1,1)$。也就是说，在前 4 个时隙内每个接收者都能监听到 3 次传输。最优调度策略应在前 3 个时隙分别发送 $\boldsymbol{P}_1$、$\boldsymbol{P}_2$、$\boldsymbol{P}_3$，那么在第 4 个时隙第 i 个接收者需要接收第 i 个数据包。显然，能够在第 4 个时隙同时完成 3 个接收者的文件下载的调度策略是不存在的，因为每个接收者丢失的数据包都不一样。然而采用编码策略时，我们可以传输 $\boldsymbol{P}_1$、$\boldsymbol{P}_2$、$\boldsymbol{P}_3$，$(\boldsymbol{P}_1+\boldsymbol{P}_2+\boldsymbol{P}_3)$（见表 9-1）从而能够在 4 个时隙内完成 3 个接收者的文件下载。不难看出在任意的信道实现下，编码所需的时隙都不会超过调度。因此，通过编码的确实现了更短的完成时间。

表 9-1　对例 9-1 的说明：R_i 表示第 i 个接收者，"－"表示信道为 OFF 状态，$a \mid b$ 分别给出了调度 | 编码策略下的最优传输。使用调度策略时，在给定的信道实现下，第 4 个时隙不论选择 $\{\boldsymbol{P}_i\}$ 中的哪个发送，都无法同时完成所有接收者的文件下载。

	$t=1$	$t=2$	$t=3$	$t=4$
R_1		$\boldsymbol{P}_2 \mid \boldsymbol{P}_2$	$\boldsymbol{P}_3 \mid \boldsymbol{P}_3$	$? \mid (\boldsymbol{P}_1+\boldsymbol{P}_2+\boldsymbol{P}_3)$
R_2	$\boldsymbol{P}_1 \mid \boldsymbol{P}_1$		$\boldsymbol{P}_3 \mid \boldsymbol{P}_3$	$? \mid (\boldsymbol{P}_1+\boldsymbol{P}_2+\boldsymbol{P}_3)$
R_3	$\boldsymbol{P}_1 \mid \boldsymbol{P}_1^2$	$\boldsymbol{P}_2 \mid \boldsymbol{P}_2$	—	$? \mid (\boldsymbol{P}_1+\boldsymbol{P}_2+\boldsymbol{P}_3)$

例 9-1 说明对于某些信道实现，网络编码相比于任意调度策略都能获得严格的性能提升。然而，目前我们还不清楚当放大或缩小系统时，上述优势是否能够显著地降低期望完成时间，参考文献（［10，12，24］）都对这个问题进行了研究。本章中我们将给出采用下述随机网络编码（RNC）策略时，关于一批数量为 K 的数据包的渐近期望完成时间的主要研究结果。

随机网络编码（**RNC**）：

While（全部 N 个接收者都没有解出全部 K 个数据包）

　　对于每个 k 都从 $\mathbb{F}_q$ 中均匀随机地选取 $a_k[t]$；

　　发送数据包 $\boldsymbol{P}[t]=\sum_{K=1}^{K}a_k[t]\boldsymbol{P}_k$；

　　$t\leftarrow t+1$；

⊖　注意，当 $N=2$ 时，如果发送者在开始传输前就有 CSI 信息，则它可以设计一种与网络编码策略性能相同的调度策略。但是我们应该看到网络编码在没有 CSI 时也可以逼近地实现相同的性能。

定理9-1　令 Z^{RNC}（N，K，$\boldsymbol{c}$）表示发送者使用RNC将一批数量为K的数据包通过一组独立时变的消失信道发送给N个接收者所需的完成时间，并定义期望向量 $\boldsymbol{c} \triangleq (c_n)_{n=1}^{N}$。那么，当域的大小$q$趋于无穷大时，上述RNC算法渐近最优，从某种意义上说也就是期望完成时间在所有的策略中渐近最小。该期望完成时间收敛于：

$$E[Z^{\mathrm{RNC}}(N,K,\boldsymbol{c})] = K + \sum_{t=K}^{\infty}\left[1 - \prod_{i=1}^{N}\left(\sum_{j=K}^{t}\binom{j-1}{K-1}(1-c_i)^{j-K}c_i^K\right)\right]$$

此外，在对称信道条件下，即$\forall n=1$，…，N都有 $c_n = c \in (0, 1)$ 时，对于固定的K和递增的N，$Z^{\mathrm{RNC}}(N, K, \boldsymbol{c})$ 的一阶矩和二阶矩分别可以表示为以下形式[11]：

$$E[Z^{\mathrm{RNC}}(N,K,\boldsymbol{c})] = \mathrm{lc}(T) + \frac{1}{2} - \frac{\gamma}{\log(1-c)} + h(\mathrm{lc}(T)) + o(1)$$

$$E[(Z^{\mathrm{RNC}}(N,K,\boldsymbol{c}))^2] = \mathrm{lc}^2(T) + \mathrm{lc}(T)\ (1+2\gamma+2g_1(\mathrm{lc}(T))) + \frac{2}{3}$$

$$-\frac{\gamma}{\log(1-c)} - \frac{(\gamma^2+(\pi^2/6))}{\log^2(1-c)} + O((\mathrm{K}-1)\mathrm{lc}(\mathrm{lc}(N))) + h(\mathrm{lc}(T)) + g_2(\mathrm{lc}(T)) + o(1)$$

其中，lc（·）表示 $\log_{\frac{1}{1-c}}(\cdot)$ 的缩写：$T = N\left(\frac{c}{(1-c)}\right)^{K-1}\frac{\mathrm{lc}^{(K-1)}N}{(K-1)!}$；$\gamma$ 是欧拉（Euler - Mascheroni）常数（约等于0.5572）。$h(\cdot)$ 是一个周期为1且期望为0的 C^{∞} - 周期函数⊖，对于 $k=0, 1, \cdots$，其傅里叶（Fourier）系数 $h_k = \frac{1}{\log(c)}\Gamma\left(\frac{2ik\pi}{\log(c)}\right)$，其中 $\Gamma(\cdot)$ 是标准伽玛函数（Gamma Function）。类似地，$g_1(\cdot)$ 和 $g_2(\cdot)$ 是两个周期为1且期望为0的 C^{∞} - 周期函数。

关键结果是两个简单的渐近式：$E[Z^{\mathrm{RNC}}(N,K,c)] \approx \mathrm{lc}N$ 以及 $E[(Z^{\mathrm{RNC}}(N,K,c))^2] \approx \mathrm{lc}^2N$，其中（$K$，$c$）固定，$N$递增。根据詹恩逊不等式（Jenson' Sinequality）有 $E[(Z^{\mathrm{RNC}}(N,K,c))^2] \geqslant E[(Z^{\mathrm{RNC}}(N,K,c))]^2$，这意味着当$N$趋于无穷大时，RNC的渐近二阶矩最优。

在更新的参考文献［33］中，作者利用极值理论也研究了完成时间分布的收敛性。极值理论研究的是形式为 $Z_N = \max_{1 \leqslant n \leqslant N} Y_n$ 的随机变量的特性，其中 Y_n 是独立同分布（i. i. d.）的随机变量。该理论表明，在一定的假设条件下，当N趋于无穷大时，Z_N 的分布将收敛于3个可能的极值分布其中之一，它们分别是Frechet、Gumbel和Weibull[5,25]。

在RNC操作下，如果把 Y_n 定义为第n个信道恰好出现K次ON状态所需的时隙数量，那么 Z_N 则对应之前定义的N个接收者接收K个数据包的完成时间 Z^{RNC}（N，K，c）。容易看出，在期望为c的i. i. d. 伯努利信道过程下，Y_n 是服从次数为K且成功概率为c的负二项分布或帕斯卡分布的随机变量。以下是参考文献［33］中给出的利用极

⊖ C^{∞} - 周期函数 $h(x)$ 是一个无限次连续可微函数，对于任意x都存在一个常数（周期）P，满足 $h(x+P)=h(x)$。

值理论得到的渐近收敛结果。

定理 9-2 对于固定的 K，在 $\forall n$ 都有 $c_n = c$ 的对称信道条件下，使用 RNC 型广播策略将 K 个数据包广播给 N 个节点所需的完成时间 $Z^{\mathrm{RNC}}(N,K,\boldsymbol{c}) =: Z(N)$ 以 Gumbel 收敛域中的随机变量为界[5,25]。即存在满足以下条件的 $Z_l(N)$ 和 $Z_u(N) = Z_l(N)+1$：

(i)

$$Z_l(N) \leqslant_{st} Z(N) \leqslant_{st} Z_u(N)$$

其中随机变量 U 和 V 满足 $U \leqslant_{st} V$ 意味着 $\forall a$ 都有 $P(U>a) \geqslant P(V>a)$；

(ii)

$$\sum_{N\to\infty} P\left(\frac{Z_l(N)-b_N}{a_N} \leqslant x\right) = \exp(-e^{-x}),\ \forall x \in \mathbb{R} \tag{9-2}$$

其中，$a_N = -1/\log(c)$，

$b_N = \mathrm{lc}(N) + (K-1)\ (\mathrm{lc}(\zeta) + \mathrm{lc}((1-c)/c)) - \mathrm{lc}\ ((K-1)!)$，

$\zeta = \mathrm{lc}(N) + (K-1)\mathrm{lc}\ ((1-c)/c)$

该结果表明完成时间的累积分布在 b_N 附近有一个明显的从 1 到 0 的转折点，这可以从式（9-2）中的极限双指数 Gumbel 型分布中看出。这一结果通过另一个渐近证实了期望完成时间随 $\mathrm{lc}(N)$ 的增大（也就是随 N 的增大）而增加。我们注意到参考文献［33］中还对更具一般性的非对称信道状态过程进行了研究。

所有的这些工作关注的都是最小化传输有限大小的数据所需的时间，因此它们都可以归入无速率传输的研究范围。参考文献［12，24］中研究了一个与无速率传输场景直接相关的指标，称为可靠性增益，它测量的是平均每个数据包所需的期望传输次数。如果对 K 个数据包使用网络编码策略，则可以用整个数据块的完成时间除以 K 来计算该指标。下节将概述关于可靠性增益指标的一些研究结果。

9.2.2 扩展

9.2.2.1 拓扑扩展

我们可以从几个方面把单跳网络的基本构件扩展为更一般的拓扑结构。本小节中我们将给出两种已经在参考文献中提出过的拓扑扩展。参考文献［12］中引入并研究了第一种拓扑结构，它是一个以源节点为根，高度为 h 的满 N 叉树，其中源节点的目标是通过独立衰落的消失链路给该树的 N^h 个叶子节点发送一组数量为 K 的数据包，每条链路处于 ON 状态的概率都是 $c \in (0,1)$（见图 9-2）。下面的定理总结了关于在上一节的最后介绍的可靠性增益指标的主要研究结果（读者可以参阅参考文献［12］进一步了解细节）。

定理 9-3 对于图 9-2 中的高度为 h 的树形拓扑，比较以下 4 种方案：(i) 端到端的 ARQ 方案（e2e－ARQ）：树的根节点一直发送数据包，直至所有位于叶子处的接收者都接收到了全部的数据包。中继节点每收到一个数据包就转发一次。(ii) 链路到链路的 ARQ 方案（ll－ARQ）：根节点和所有中继节点需要确保它们的所有子节点都成功接收到了数据包，因此对于每个数据包，中继节点都有可能进行多次传输。(iii) 端到端的 FEC

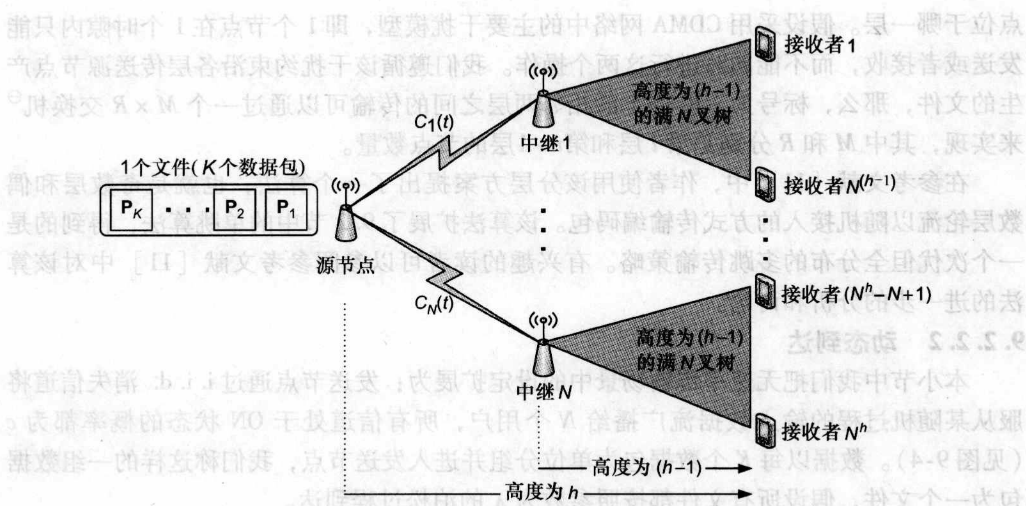


图 9-2　将图 9-1 扩展为一棵高度为 h 的满 N 叉树，其中每条消失链路的衰落都相互独立，期望值均为 c

方案（e2e－FEC）：该方案允许根节点进行分组编码来恢复一些失败的传输，中继节点仅转发数据包而不进行进一步处理。(iv) 随机网络编码（RNC）：每个中继节点在成功解码得到来自其父节点的 K 个数据包之后，都会对这 K 个数据包进行随机网络编码（具体参考 9.2 节中定义的 RNC 算法），以确保它的所有子节点都能成功接收到它的数据包。

不同方案对于网络编码方案的相对可靠性增益 ρ 定义为该方案的可靠性增益（每个数据包的传输次数）与网络编码方案（上述第 4 种方案）的可靠性增益的比值。那么，对于固定的 K 和递增的 N，以下标度律成立：

- $\rho^{(e2e-ARQ)}(N,c,h)=\Theta(\log N)$；
- $\rho^{(ll-ARQ)}(N,c,h)=\Theta(c^{-(h-1)}\log N)$；
- $\rho^{(e2e-FEC)}(N,c,h)=\Theta(c^{-h})$。

定理 9-3 表明传输次数的相对增益随网络大小以及树的高度的增加而增大。

参考文献［11］中介绍了另一种拓扑扩展，它把一般的拓扑结构重排为分层拓扑，并把分层拓扑看做一连串的单跳网络来分析，因此可以将上述单跳模型应用到多跳无线网络中。下面举例说明该分层方案。

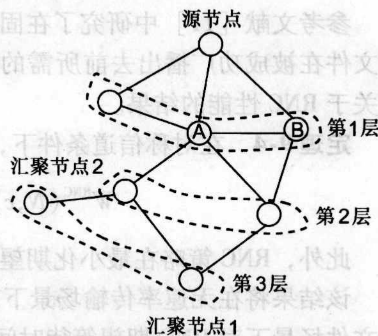


图 9-3　一般网络拓扑下的一个多播设定

例 9-2（按层分解网络）　考虑图 9-3 所示的包含两个汇聚节点、一个源节点和一些中继节点的多播设定。

我们按层对该网络进行分解：如果一个节点到源节点的最短路径长度为 i 跳，那么该节点就属于第 i 层。我们可以通过简单地洪泛网络或使用最短路径算法来确定每个节

点位于哪一层。假设采用 CDMA 网络中的主要干扰模型，即 1 个节点在 1 个时隙内只能发送或者接收，而不能同时进行这两个操作。我们遵循该干扰约束沿各层传送源节点产生的文件，那么，标号为 i 和 $i+1$ 的相邻两层之间的传输可以通过一个 $M\times R$ 交换机[⊖]来实现，其中 M 和 R 分别是第 i 层和第$i+1$层的节点数量。

在参考文献［11］中，作者使用该分层方案提出了一个算法，也就是奇数层和偶数层轮流以随机接入的方式传输编码包。该算法扩展了 9.2 节中的单跳算法，得到的是一个次优但全分布的多跳传输策略。有兴趣的读者可以参阅参考文献［11］中对该算法的进一步的分析和讨论。

9.2.2.2　动态到达

本小节中我们把无速率传输场景中的设定扩展为：发送节点通过 i. i. d. 消失信道将服从某随机过程的输入数据流广播给 N 个用户，所有信道处于 ON 状态的概率都为 c（见图 9-4）。数据以每 K 个数据包为单位分组并进入发送节点，我们称这样的一组数据包为一个文件，假设所有文件都按照参数为 λ 的泊松过程到达。

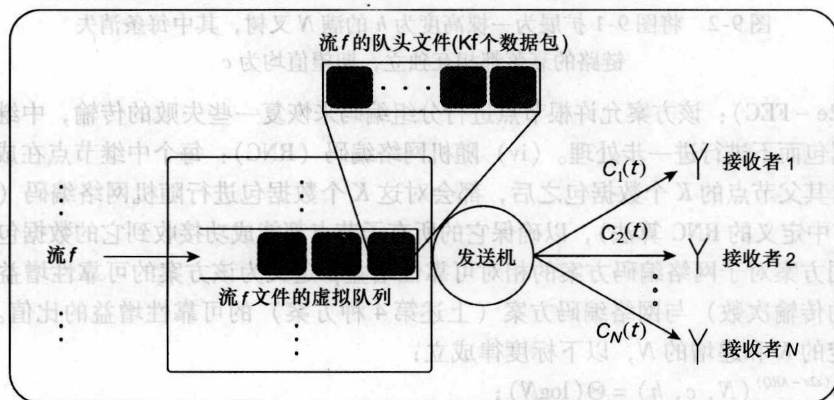


图 9-4　将无速率传输场景中的设定扩展为数据流，其中 HOL 表示队头

参考文献［11］中研究了在固定的文件大小 K 和递增的用户数量 N 下，进入队列的文件在被成功广播出去前所需的等待时间的指标 $W(N, c)$。通过该研究可以得到以下关于 RNC 性能的结果。

定理 9-4　在对称信道条件下，对于固定的 K 和可变的 N，RNC 的等待时间性能为

$$W^{\mathrm{RNC}}(N,c)=\Theta\left(\frac{\lambda \mathrm{l}c^2N}{2(1-\lambda \mathrm{l}cN)}\right),\lambda<\frac{1}{\mathrm{l}cN}$$

此外，RNC 策略在最小化期望等待时间方面对 N 渐近最优。

该结果将在无速率传输场景下对 RNC 期望完成时间的最优化扩展为在发送端广播流文件场景下对 RNC 期望等待时间的最优化。这种延伸建立在定理 9-1 所阐述的 RNC 渐近二阶矩最优的基础上。然而需要注意的是，上述结果成立的前提是假设 K 固定，

⊖　一个 $M\times R$ 网络交换机可以将到达它的 M 个输入端口的数据包流转发给相应的 R 个输出端口，通常可以用一个二分图来表示，二分图中的有向链路从每个输入端口指向每个输出端口。

因此随着 N 的增大，可支持速率 λ 会和 $\frac{1}{\ln N}$ 一样减小。在 9.2.3 节中我们将讨论对 K 缩放产生的效果。

9.2.2.3　流入流量的延迟灵敏度

参考文献 [3] 将图 9-4 中的网络设定扩展为图 9-5 所示的场景。其中与前者相同的是文件的到达服从速率为 γ 的泊松过程，不同的是添加了一个相关的测量其代价值的独立生成的随机变量 U。此外，接收者需要给每个发送该文件给它的发送者都支付代价 p。变量 U 测量的是该文件能够接受的总等待时间与总传输代价之和。当 U 值较小的文件进入系统时，它们会被概率性地丢弃，因此被接收的文件都服从速率 λ 较小的泊松过程。

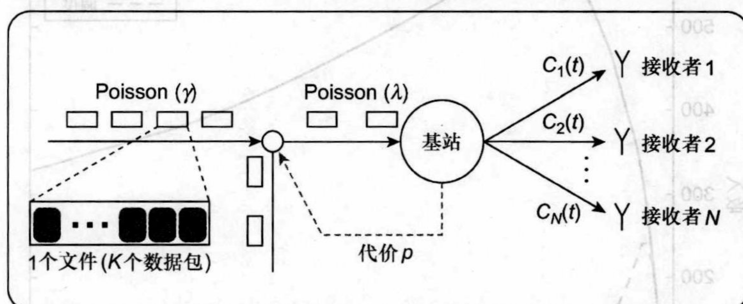


图 9-5　对于延迟灵敏度的场景扩展

显然，随着每个传输的代价 p 增大，被接收的数据包比例将减少。而另一方面，当 p 减小时，虽然被接收的数据包比例将增大，但其等待时间也会增加。参考文献 [3] 中研究了当改变 N 和 K 时，如何设定代价值才能使收入最大化的问题。图 9-6 中所示

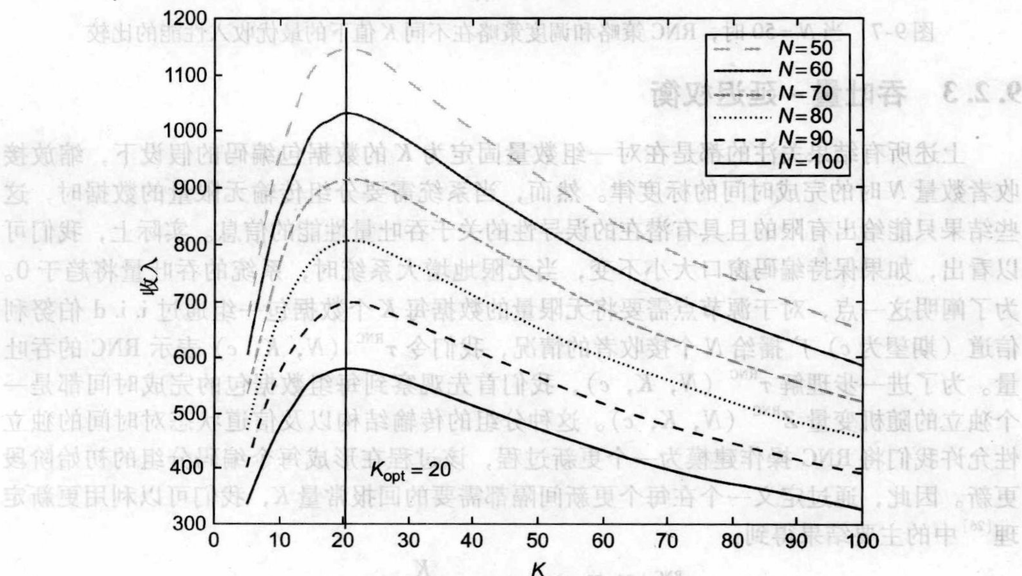


图 9-6　RNC 策略在不同 N 值和 K 值下的最优收入性能

的是在 RNC 传输策略下，最大收入性能作为 K 和 N 的函数的变化情况。从该图中我们观察到了一个有趣的现象：最优收入水平作为 K 的函数具有单峰性质。此外我们还观察到，最优的 K_{opt} 的选择对于网络大小的变化极其不敏感。这个现象非常引人注目，因为它意味着最优的编码窗口大小和用户数量之间的自然去耦。

图 9-7 中所示的是对于不同的 K 和固定的 N，在没有 CSI 可利用的情况下，RNC 策略与最优调度策略的性能比较，即网络编码策略相比于传统调度策略的相对增益。从图中可以看出，网络编码传输策略实现了对收入增益的数量级的提升。

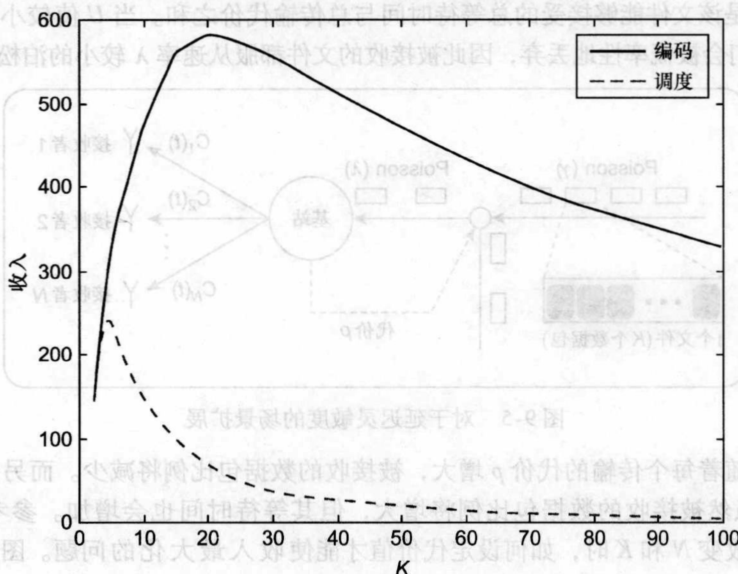


图 9-7　当 $N=50$ 时，RNC 策略和调度策略在不同 K 值下的最优收入性能的比较

9.2.3　吞吐量－延迟权衡

上述所有结果关注的都是在对一组数量固定为 K 的数据包编码的假设下，缩放接收者数量 N 时的完成时间的标度律。然而，当系统需要分组传输无限量的数据时，这些结果只能给出有限的且具有潜在的误导性的关于吞吐量性能的信息。实际上，我们可以看出，如果保持编码窗口大小不变，当无限地增大系统时，系统的吞吐量将趋于 0。为了阐明这一点，对于源节点需要将无限量的数据每 K 个数据包一组通过 i. i. d 伯努利信道（期望为 c）广播给 N 个接收者的情况，我们令 $\tau^{RNC}(N, K, c)$ 表示 RNC 的吞吐量。为了进一步理解 $\tau^{RNC}(N, K, c)$，我们首先观察到每组数据包的完成时间都是一个独立的随机变量 $Z^{RNC}(N, K, c)$。这种分组的传输结构以及信道状态对时间的独立性允许我们将 RNC 操作建模为一个更新过程，该过程在形成每个编码分组的初始阶段更新。因此，通过定义一个在每个更新间隔都需要的回报常量 K，我们可以利用更新定理[26]中的主要结果得到：

$$\tau^{RNC}(N,K,c)=\frac{K}{E[Z^{RNC}(N,K,c)]}$$

显然，对于任意固定的K，当N趋于∞时，由于$E[Z^{\text{RNC}}(N,K,c)]$随$\text{lc}(N)$的增大而增大（参见定理9-1），因此$\tau^{\text{RNC}}(N, K, c)$将趋近于0。这意味着如果我们增大网络的大小$N$，为了使吞吐量非零，分组大小$K$也必须随之增大。然而，增大$K$意味着延长了每个分组的传输持续时间。因此，我们有必要研究能够实现最优吞吐量－延迟权衡的K和N。

首先我们注意到对于任何策略π都有$\tau^{\pi}(N,K,c)\leqslant c$，这是因为右边的值表示平均每个接收者成功接收一个数据包所需的时间。仅当几乎所有接收到的数据包都是有效数据包时，才能达到这个上界。有趣的是，有研究证明，当K和N确定时，使用RNC能够实现该上界。下面的例子突出强调了缩放编码窗口的大小对RNC的吞吐量－延迟性能的影响，并带动了后续的研究。

例9-3 考虑以下场景：一个单独的源节点使用RNC策略将一组数量为K的数据包通过期望为c的i. i. d消失信道广播给N个接收者。对于大小为d的足够大的域来说，如果使用标准的随机线性编码理论（参见参考文献［15］），接收者只要接收到K个或略多于K个编码数据包就足以解码这一组数据。

每个接收者都至少需要K个时隙才能解码一整组数据。令随机变量$M[t]$表示在$t\geqslant K$个时隙内成功解码出了K个数据包的接收者的数量，令$r[t]$表示任意给定的一个接收者在$t\geqslant K$个时隙内接收到至少K个数据包的概率，则$M[t]$是一个成功概率为$r[t]$的二项随机变量，其中$r[t]=\sum_{l=K}^{t}\binom{t}{l}c^{l}(1-c)^{t-l}$，且有$E(M[t])=Nr[t]$。因此$r[t]$也表示在$t$个时隙内成功解码出$K$个数据包的接收者所占的比例。

为了比较$r[t]$作为t的函数在不同K值下的表现，我们定义了一个归一化的时间变量$s=\frac{t-K}{K}$。相应地，我们还定义了变量$r'[s]=r[Ks+K]$，它可以看做是在s个时隙内成功解码出数量为K的一组包中的一个数据包的接收者比例。通过比较不同K值下的$r'[s]$，我们可以看到在一个归一化的时间尺度内从数量为K的一组数据包中解码出一个等价数据包的接收者的比例。

我们对$r'[s]$进行了数值计算，当$c=0.5$时，不同K值下$r'[s]$随s的变化如图9-8所示。从图中可以看出，当$K=30$时，源节点在一段较短的时间内为大部分用户提供了服务，但却花费了相对较长的一段时间来为余下的小部分用户服务，以完成对当前该组K个数据包的传输。当我们将K增大到60，进而增大到120的时候，曲线的过渡部分变得越来越急剧，这表明了源节点在更短的时间内为更大一部分用户提供了服务，而花费了更少的时间为余下的更小一部分用户服务，以完成对当前组的传输。为了增大吞吐量，我们理想地希望所有的用户都能够同时完成解码。正如在图9-8中所看到的，我们可以通过无限地增大K来实现这一点。然而，这同时也会导致解码延迟的无限增加。因此在不同的N值下，理解当K变化时吞吐量和延迟之间的折衷关系是非常重要的。

在参考文献［30］中，作者研究了上述问题，通过适当地对关键系统参数进行高斯近似，研究了RNC的吞吐量和延迟性能，并得到了关于其表现的近似紧界。特别要

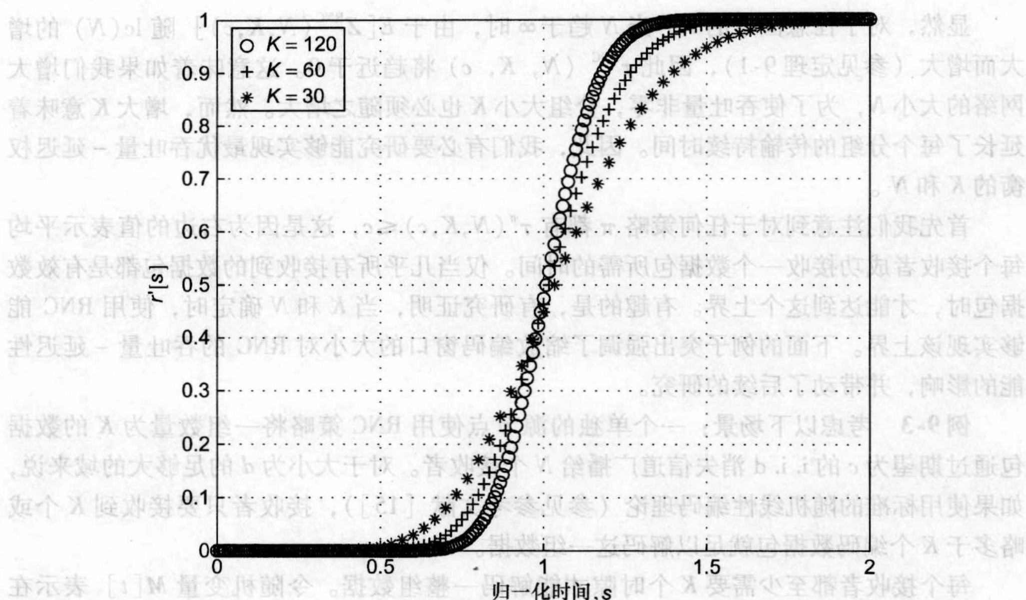


图 9-8　当 $c=\frac{1}{2}$时，在 s 个时隙内成功解码出数量为 K 的一组数据包中的 1 个的接收者比例 $r'[s]$ 随 s 的变化情况

指出的是，该研究表明，编码窗口大小（也就是解码延迟）$K=\Theta(\ln(N))$ 代表了一个反转速率，当速率低于该值时，吞吐量收敛于零；当速率高于该值时，吞吐量收敛于广播容量 c（即能够通过消失信道把数据广播给全部 N 个用户的最大速率）。在给出该结果之前，我们先描述上面提到的高斯近似。

定义 9-3　定义解码延迟的近似值 $\tilde{Z}(N, K)$ 和期望吞吐量的近似值 $\tilde{\tau}^{\mathrm{RNC}}(N, K)$ 如下：

$$\tilde{Z}^{(RNG)}(N,K)=\frac{K}{c}+\frac{\sqrt{K(1-c)}}{c}\max_{1\le i\le N}\chi_i \tag{9-3}$$

$$\tilde{\tau}^{\mathrm{RNC}}(N,K)=\frac{K}{\mathbb{E}[\tilde{Z}^{(RNG)}(N,K)]} \tag{9-4}$$

其中，$\tilde{\chi}_i$，$i=1,2,\cdots,N$ 是相互独立的标准正态随机变量。

定理 9-5　期望吞吐量的近似值 $\tilde{\tau}^{\mathrm{RNC}}(N, K)$ 具有如下性质：

1）如果 $K=o(\ln(N))$，那么当 $N\to\infty$ 时，$\tilde{\tau}^{\mathrm{RNC}}(N, K)$ 趋于 0。

2）如果 $K=\Theta(\ln(N))$，那么当 $N\to\infty$ 时，$\tilde{\tau}^{\mathrm{RNC}}(N, K)$ 趋于广播容量 c。

3）如果 $K=\omega(\ln(N))$，那么当 $N\to\infty$ 时，$\tilde{\tau}^{\mathrm{RNC}}(N, K)$ 趋于 c。

图 9-9 和 9-11 在 $c=0.9$ 的 i. i. d. 消失信道下证实了上述定理中给出的预测。其中图 9-9 通过研究在不同的 K 和 N 的比例关系下的吞吐量性能，证实了其反转性质。正如高斯近似中指出的，$K=\Theta(\log(N))$ 代表了一个出现急剧反转的标度，高于该标度则吞吐量可以近似达到其最大可能值 $c=0.9$，低于该标度则吞吐量将渐近于 0。

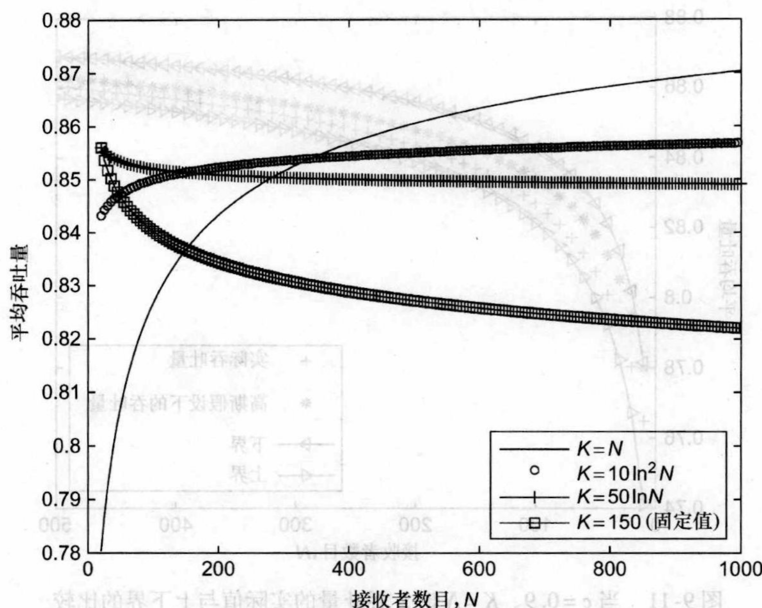


图 9-9　在不同的 K 与 N 的比例关系下的期望吞吐量性能

图 9-10 和 9-11 表明了实际的系统性能与上文给出的高斯近似十分接近，并检验了所求的近似界对于实际性能的有效性。这些数值研究不仅证实了理论近似结果的有效性，还让我们看到了所求的近似上界和下界都是十分紧的。最近的研究工作[31]中推导了实际系统性能的紧上界和紧下界，从而证明了定理 9-5 中给出的基于上述高斯近似的渐近结果对于实际系统来说是非常精确的。

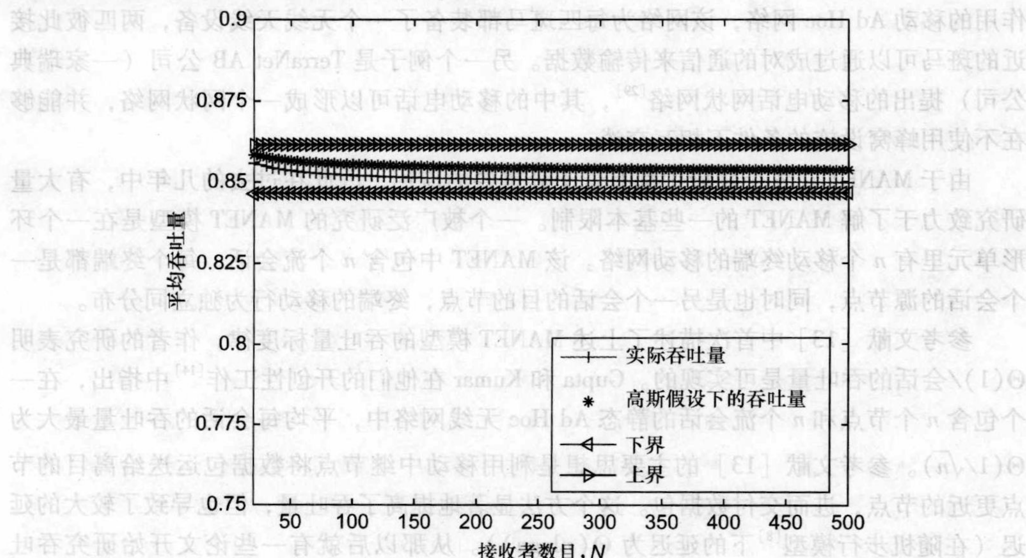


图 9-10　当 $c=0.9$、$K=50\ln(N)$ 时，吞吐量的实际值与上下界的比较

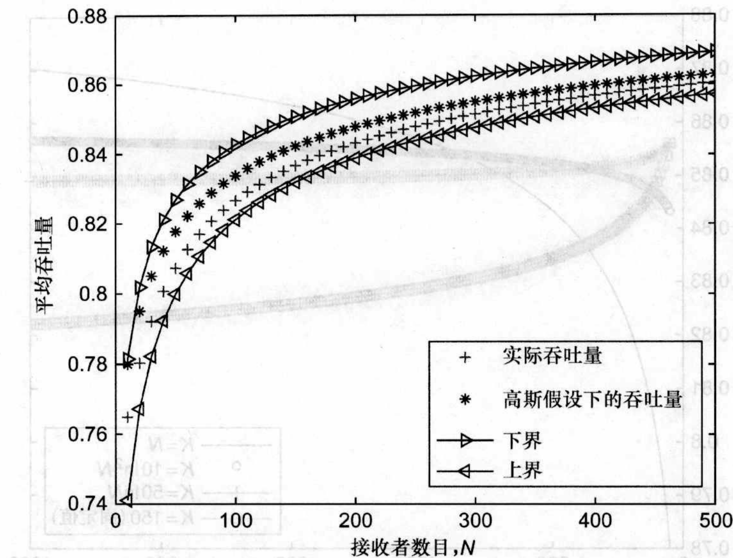


图 9-11 当 $c=0.9$、$K=N$ 时，吞吐量的实际值与上下界的比较

9.3 大规模移动 Ad Hoc 网络中的编码

网络编码的另一个重要应用是在移动 Ad Hoc 网络（MANET）中。MANET 是最具创新性的新兴网络技术之一，在个人区域网络、紧急与救援行动以及军事战场等方面有着潜在的广泛应用。例如，ZebraNet[36] 是一个用于监测和研究动物的迁移以及种间相互作用的移动 Ad Hoc 网络，该网络为每匹斑马都装备了一个无线天线设备，两匹彼此接近的斑马可以通过成对的通信来传输数据。另一个例子是 TerraNet AB 公司（一家瑞典公司）提出的移动电话网状网络[29]，其中的移动电话可以形成一个网状网络，并能够在不使用蜂窝设施的条件下相互交谈。

由于 MANET 在各个领域中都有着十分重要的应用，因此在过去的几年中，有大量研究致力于了解 MANET 的一些基本限制。一个被广泛研究的 MANET 模型是在一个环形单元里有 n 个移动终端的移动网络。该 MANET 中包含 n 个流会话，每个终端都是一个会话的源节点，同时也是另一个会话的目的节点，终端的移动行为独立同分布。

参考文献［13］中首次描述了上述 MANET 模型的吞吐量标度律，作者的研究表明 $\Theta(1)$/会话的吞吐量是可实现的。Gupta 和 Kumar 在他们的开创性工作[14] 中指出，在一个包含 n 个节点和 n 个流会话的静态 Ad Hoc 无线网络中，平均每会话的吞吐量最大为 $\Theta(1/\sqrt{n})$。参考文献［13］的主要思想是利用移动中继节点将数据包运送给离目的节点更近的节点，进而交付数据包。这个方法显著地提高了吞吐量，但也导致了较大的延迟（在随机步行模型[8] 下的延迟为 $\Theta(n\log n)$）。从那以后就有一些论文开始研究吞吐量和延迟之间的权衡关系[4,6-9,13,20-23,28,32]。在这些论文中一个被广泛使用的方法是将一

个数据包转发给多个终端，当这些终端中的某一个靠近目的节点时，该数据包就能够被送达。这个方法显著地降低了平均每个数据包的传输时间（延迟），但是该方法对数据包的复制操作会导致吞吐量的降低。

在最近的研究工作[34]中，作者提出将线性网络编码应用于 MANET。相比于不进行编码的算法来说，这些编码算法能够提升系统性能的数量级。下面是参考文献［34］中给出的一个例子，用于说明在 MANET 中使用编码的关键之处。

9.3.1 示例：在 i. i. d 移动模型下的吞吐量 - 延迟权衡

考虑在一个环形单元中部署了 n 个移动终端的移动网络。假设节点 i 发送信息给节点 $(i+1)\mathrm{mod}(n)$，因此网络中有 n 个数据流。节点 i 和节点 $(i+1)\mathrm{mod}(n)$ 称为一个源 - 目的节点对。终端的移动服从一个 i. i. d. 移动模型。

定义 9-4（i. i. d. 移动模型） n 个无线移动节点随机均匀地分布在一个环形单元中。不同节点的位置彼此独立，同一节点的位置在不同时隙彼此独立。因此，在该 i. i. d. 移动模型下，所有终端的位置在每个时隙都会被重新洗牌。

此外，假设我们可以使用参考文献［1］中引入的协议模型来对无线干扰建模。

定义 9-5（协议模型[1]） 令 α_i 表示节点 i 的传输半径，那么当且仅当以下两个条件同时成立时，该协议才认为从节点 i 到节点 j 的一次传输是成功的：(i) 节点 i 和节点 j 之间的距离小于 α_i；(ii) 如果在同一时刻终端 k 发送信息给终端 h，那么节点 k 和节点 j 之间的距离应至少为 $(1+\Delta)\ \alpha_k$，其中 $\Delta>0$ 定义了一个该传输的保护区域。

限制 MANET 性能的基本约束有两个：

- 无线干扰限制了每个时隙同时进行的传输数量。
- 移动性决定了一个数据包到达目的节点途经的距离。

乍看起来，这两个约束条件似乎完全不同，这使得对 MANET 的容量描述成为了一个非常有挑战性的问题。参考文献［34］的作者使用一个虚拟信道框架统一了这两个约束，从而克服了这一障碍。该虚拟信道框架展示了线性网络编码在 MANET 中的关键作用。

为了简化分析，假设我们使用参考文献［13］中提出的两跳传输算法来传输数据包。

定义 9-6（两跳传输算法） 两跳传输算法的工作原理如下：

- 第一跳：将数据包从其源节点发送给该源节点周围的一个（或若干个）中继节点；
- 第二跳：当某个中继节点距离数据包的目的节点足够近时，该中继节点就将该数据包发送给目的节点。

上述两跳传输算法不支持多跳中继，但该算法把握了成功交付一个数据包的 3 个关键阶段（见图 9-12）：

- 第Ⅰ阶段：源节点将数据包发送给一个或多个中继节点，假设传输半径是 L_1；
- 第Ⅱ阶段：某个中继节点移动到该数据包的目的节点附近的位置；
- 第Ⅲ阶段：该中继节点将该数据包发送给目的节点，假设该阶段的传输半径是 L_2。

在参考文献［34］中，作者把每个阶段都建模为一个虚拟通信信道：

- **可靠的广播信道**：不失一般性地，我们假设 $\Delta=2$。在上述协议模型下，网络中最多有 $\frac{1}{\pi(L_1)^2}$ 个传输能够同时进行。平均每个终端可用于将其数据包广播给中继节点的时间为

$$P_1=\frac{1}{\pi(L_1)^2 n}$$

平均每个数据包将会被转发给 $\pi(L_1)^2 n$ 个移动终端。一个虚拟信道的容量为

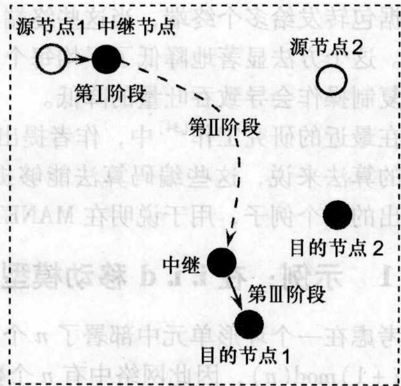


图 9-12　在两跳传输方案下，一次典型传输的 3 个阶段，虚线表示中继节点在一个或多个时隙内的移动轨迹

$$\frac{W}{\pi(L_1)^2 n}$$

其中，W 是该信道的一个移动终端在一个时隙内能够发送给另一个移动终端的数据包的最大数量。

- **不可靠的中继信道**（消失信道）：如果一个中继节点与某数据包的目的节点之间的距离小于 L_2，那么它就可以将该数据包转发给该目的节点。假如上述事件发生，我们就说我们转发的这个数据包是可送达的。在 i. i. d. 移动模型下，某个数据包可送达的概率为

$$\pi(L_2)^2$$

前面提到，平均每个数据包会被转发给 $\pi(L_1)^2 n$ 个移动终端。在 D 个连续的时隙内，它们全部都不可送达的概率为

$$(1-\pi(L_2)^2)^{\pi(L_1)^2 nD}$$

因此，我们可以用消失信道来对中继阶段建模，其消失概率为

$$P_{\text{erasure}}=(1-\pi(L_2)^2)^{\pi(L_1)^2 nD}$$

- **可靠的接收信道**：考虑从中继节点到目的节点的传输。在上述协议模型下，同一时隙内成功送达的数据包数量不会超过：

$$\frac{1}{\pi(L_2)^2}$$

平均每个目的节点用于接收数据包的时间为

$$\frac{1}{\pi(L_2)^2 n}$$

因此该接收信道的容量为

$$\frac{W}{\pi(L_2)^2 n}$$

基于以上虚拟信道参数，我们可以用图 9-13 所示的 3 个虚拟信道的串联来表示 MANET。基于该实验性推理，我们可以计算出平均每个会话的最大吞吐量为

$$\lambda = \max_{L_1,L_2}\min\left\{(1-(1-\pi\ (L_2)^2)^{\pi D(L_1)^2_n})\ \frac{W}{\pi n\ (L_1)^2},\ \frac{W}{n\pi\ (L_2)^2}\right\}$$
$$= \Theta\left(\sqrt{\frac{D}{n}}\right)$$

其中，使等式的右边取最大值的传输半径 L_1 和 L_2 分别是：$L_1^* = \Theta\left(\frac{1}{\sqrt[2]{n}}\right)$和 $L_2^* = \Theta\left(\frac{1}{\sqrt[4]{Dn}}\right)$。显然，要实现这个吞吐量需要一个可以实现消失信道容量的编码方案。考虑到消失概率取决于 L 和 D，而它们在不同的延迟约束下是不同的，因此线性网络编码成为了我们的第一选择。

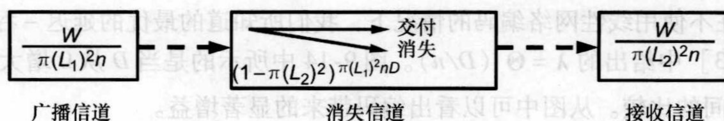


图 9-13　虚拟信道建模

要强调的是，虚拟信道不仅表现出了在 MANET 中通信的限制，还揭示了线性网络编码在 MANET 中的重要性。基于上述启发式算法，在参考文献［34］中作者提出了如下的联合编码 - 调度算法，并证明了该算法能够实现延迟和吞吐量之间的最优权衡。

联合编码 - 调度算法[34]：将一个单位正方形区域划分为若干个边长为 $1/\sqrt[4]{nD}$的正方形单元。规定所有节点的传输半径均为$\sqrt{2}/\sqrt[4]{nD}$，因此同一单元内的任意两个节点都能相互通信。这意味着，在给定的干扰约束下，当某单元的固定距离内的所有单元中的所有节点都保持沉默时，该单元中的两个节点就可以相互通信。我们把每个时隙进一步划分为 9 个微时隙，并保证在每个时隙内，每个单元至少有一个微时隙处于活动状态。之所以要划分出 9 个微时隙，是因为如果某个单元中的某个节点是活动的，那么与它相邻的 8 个单元中的全部节点都不允许处于活动状态，但该单元邻域以外的节点是可以活动的。此外，我们用 $W/18$ 表示每个数据包的大小，在每个微时隙内能够发送 2 个数据包。我们将每 $6D$ 个时隙组成一个超时隙。在每个超时隙中，节点都进行如下的数据包传输：

1）随机线性编码：每个源节点取 $6D/(25M)$ 个数据包，并使用随机线性编码产生 D/M 个编码数据包，其中 $M=\sqrt{n/D}$。

2）广播：该步骤包含 D 个时隙。节点在每个时隙都执行以下操作：

(i) 如果在第 t 个时隙，某单元中的节点数量在 $9M/10$ 和 $11M/10$ 之间，我们就说该单元在该时隙是良好的。从每个良好的单元中随机选取一个节点，如果该节点尚未发送出所有的 D/M 个编码数据包，那么在分配给该单元的微时隙内，该节点会广播一个之前没有给该单元中的其他 $9M/10$ 个节点发送过的编码数据包。前文提到，通过选择适当的数据包大小，在每个时隙内每个良好的单

元中都有一个节点可以发送。

(ii) 所有节点都检查其数据包的重复性。如果两个或更多数据包的目的节点相同，则从其中随机选择一个并丢弃其他的数据包。

3）接收：该步骤包含 $5D$ 个时隙。在每个时隙内，如果一个单元包含的可交付数据包不超过 2 个，则在该单元的微时隙内通过单跳传输将这些可交付数据包交付给它们的目的节点；否则，该单元中的节点都不尝试发送。在该步骤的最后，所有不可交付的数据包都会被丢弃。目的节点会解码接收到的编码数据包。

定理 9-6　在平面 i. i. d. 移动模型下，当给定延迟约束为 $6D$ 时，平均每个源 - 目的节点对的吞吐量是 $\lambda = O(\sqrt{D/n})$。当 D 同时满足 $\omega(\sqrt[3]{n})$ 和 $o(n)$ 时，使用联合编码 - 调度算法可以实现该吞吐量。

注意到在不使用线性网络编码的情况下，我们所知道的最优的延迟 - 吞吐量权衡是参考文献 [23] 中给出的 $\lambda = \Theta(D/n)$。图 9-14 中所示的是当 D 从 0 增大到 n 时，D/n 和 $\sqrt{D/n}$ 之间的比较。从图中可以看出编码带来的显著增益。

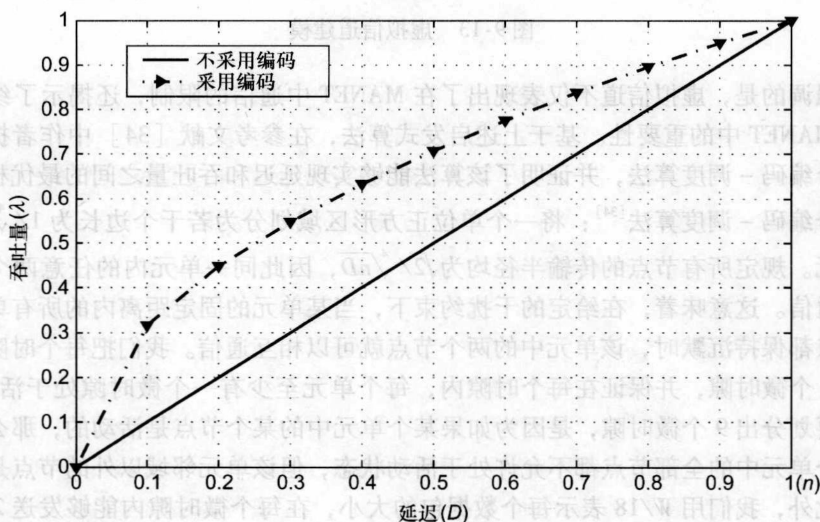


图 9-14　使用编码和不使用编码时的延迟 - 吞吐量权衡

9.3.2　对多播通信流的扩展

我们可以将上述框架扩展到多播的情况。在参考文献 [37] 中，作者考虑了包含 n_s 个多播会话的网络，其中每个多播会话都由 1 个源节点和 p 个目的节点组成。与单播场景相似，作者构建了一个如图 9-15 所示的虚拟信道系统。注意，该虚拟系统与第 9.2 节中给出的广播网络相似。因此，我们可以再次使用线性网络编码来显著地提升该多播 MANET 的性能。

我们注意到在多播情况下，同一个会话的所有目的节点都对相同的信息感兴趣。当

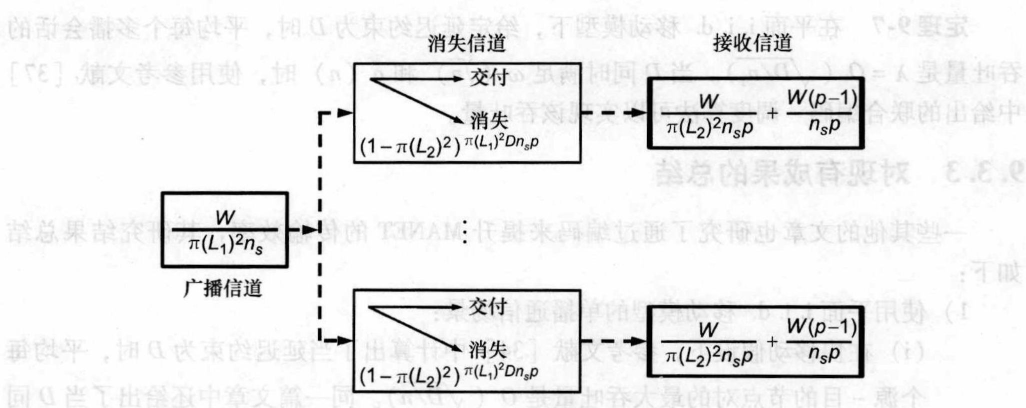


图9-15　多播会话的虚拟信道表示

我们将一个数据包交付给它的某一个目的节点时，在该传输范围内的其他目的节点也都能接收到这个数据包，从而导致多重交付。这与单播的情况是不同的。作者称这些交付中的一个为目标交付，其余为免乘交付。在该协议模型下，不同的目标交付对应的排斥区域应彼此不相交。假设共同的传输半径是 L_2，一个成功的目标交付对应一个面积为 $\pi(L_2)^2$ 的排斥区域。则同时发生的目标交付的数量不会大于：

$$\frac{W}{\pi(L_2)^2}$$

且伴随每个目标交付的免乘交付的平均数量为

$$(p-1)\pi(L_2)^2$$

因此，平均每个时隙的交付数量为

$$\frac{W[1+(p-1)\pi(L_2)2]}{\pi(L_2)^2}$$

前文提到，一个源数据包需要被交付给 p 个目的节点，因此平均每个多播会话的吞吐量为

$$W\frac{1+(p-1)\pi(L_2)^2}{n_sp\pi(L_2)^2}=\frac{W}{n_sp\pi(L_2)^2}+\frac{W(p-1)}{n_sp}$$

单位为数据包/时隙。基于虚拟信道表示，我们可以启发式地计算出：

$$\lambda=\max_{L_1,L_2}\left\{(1-(1-\pi(L_2)^2)^{\pi D(L_1)^2n})\frac{W}{\pi(L_1)^2n_s},\frac{W}{n_sp\pi(L_2)^2}+\frac{W(p-1)}{n_sp}\right\}$$
$$=\Theta\left(\sqrt{\frac{D}{n_s}}\right)$$

其中，使等式右边取最大值的传输半径 L_1 和 L_2 分别是：$L_1^*=\Theta\left(\frac{1}{\sqrt[2]{n_s}}\right)$和 $L_2^*=\Theta\left(\frac{1}{4\sqrt{p^2Dn_s}}\right)$。

定理 9-7　在平面 i. i. d. 移动模型下，给定延迟约束为 D 时，平均每个多播会话的吞吐量是 $\lambda = O(\sqrt{D/n_s})$。当 D 同时满足 $\omega(\sqrt[3]{n})$ 和 $o(n)$ 时，使用参考文献［37］中给出的联合编码 - 调度算法可以实现该吞吐量。

9.3.3　对现有成果的总结

一些其他的文章也研究了通过编码来提升 MANET 的传输效率，其研究结果总结如下：

1）使用平面 i. i. d. 移动模型的单播通信场景：

(i) 在快移动假设下，参考文献［34］中计算出了当延迟约束为 D 时，平均每个源 - 目的节点对的最大吞吐量是 $O(\sqrt{D/n})$。同一篇文章中还给出了当 D 同时满足 $\omega(\sqrt[3]{n})$ 和 $o(n)$ 时实现最大吞吐量的联合编码 - 调度算法。参考文献［35］中证明了当 $D=\Theta(n)$ 时，$\Theta(1)$ 是可实现的；当 $D=\Theta(\log n)$ 时，$\Theta(1/n)$ 是可实现的。

(ii) 在慢移动假设下，参考文献［34］中计算出了当延迟约束为 D 时，平均每个源 - 目的节点对的最大吞吐量是 $O(\sqrt[3]{D/n})$。参考文献［34］中还给出了当 D 同时满足 $\omega(1)$ 和 $o(n)$ 时实现最大吞吐量的联合编码 - 调度算法。

2）使用平面混合随机步行模型的单播通信场景：

(i) 在快移动假设下，参考文献［34］中计算出了当 $S=o(1)$ 且 $D=\omega(|\log S|/S^2)$ 时，平均每个源 - 目的节点对的最大吞吐量是 $O(\sqrt{D/n})$，其中 S 是随机步行的步长。参考文献［34］中还给出了当 $S=o(1)$ 且 D 同时满足 $\omega(\max\{(\log^2 n)|\log S|/S^6, \sqrt[3]{n}\log n\})$ 和 $o(n/\log^2 n)$ 时实现最大吞吐量的联合编码 - 调度算法。当 $S=1/\sqrt{n}$ 时，参考文献［18，35］中证明了如果 $D=\Theta(n)$，则 $\Theta(1)$ 是可实现的；参考文献［35］中证明了如果 $D=\Theta(\log n)$，则 $\Theta(1/n)$ 的吞吐量是可实现的。

(ii) 在慢移动假设下，参考文献［34］中计算出了当 $S=o(1)$ 且 $D=\omega(|\log S|/S^2)$ 时，平均每个源 - 目的节点对的最大吞吐量是 $O(\sqrt[3]{D/n})$，并给出了当 $S=o(1)$ 且 D 同时满足 $\omega((\log^2 n)|\log S|/S^6)$ 和 $o(n/\log^2 n)$ 时实现最大吞吐量的联合编码 - 调度算法。

3）使用一维 i. i. d. 移动模型的单播通信场景：

(i) 在快移动假设下，参考文献［34］中计算出了当延迟约束为 D 时，平均每个源 - 目的节点对的最大吞吐量是 $O(\sqrt[3]{D^2/n})$。参考文献［34］中还给出了当 D 同时满足 $\omega(\sqrt[5]{n})$ 和 $o(\sqrt{n}/^{3/2}\sqrt{\log n})$ 时实现最大吞吐量的联合编码 - 调度算法。

(ii) 在慢移动假设下，参考文献［34］中计算出了平均每个源 - 目的节点对的最大吞吐量是 $O(\sqrt[4]{D^2/n})$。参考文献［34］中还给出了当 D 为 $o(\sqrt{n}/\log^2 n)$

时实现最大吞吐量的联合编码 – 调度算法。

4）使用一维混合随机步行模型的单播通信场景：

（i）在快移动假设下，参考文献［34］中计算出了当 $S=o(1)$ 且 $D=\omega(1/S^2)$ 时，平均每个源 – 目的节点对的最大吞吐量是 $O(\sqrt[3]{D^2/n})$，并给出了当 $S=o(1)$ 且 D 同时满足 $\omega(\max\{(\log^2 n)|\log S|/S^4, \sqrt[5]{n}\log n\})$ 和 $o(\sqrt{n}/^{3/2}\sqrt{\log n})$ 时实现最大吞吐量的联合编码 – 调度算法。

（ii）在慢移动假设下，参考文献［34］中计算出了当 $S=o(1)$ 且 $D=\omega(1/S^2)$ 时，平均每个源 – 目的节点对的最大吞吐量是 $O(\sqrt[4]{D^2/n})$，并给出了当 $S=o(1)$ 且 D 同时满足 $\omega((\log^2 n)|\log S|/S^4)$ 和 $o(\sqrt{n}/\log^2 n)$ 时实现最大吞吐量的联合编码 – 调度算法。

5）使用平面 i. i. d. 移动模型的多播通信场景：

参考文献［37］中证明了当给定延迟约束 D 时，平均每个多播会话的吞吐量是 $O\left(\min\left\{1, (\log p)(\log n_s p)\sqrt{\frac{D}{n_s}}\right\}\right)$。同一篇文章中还给出了实现大小为 $\Theta\left(\min\left\{1, \sqrt{\frac{D}{n_s}}\right\}\right)$ 的吞吐量的联合编码 – 调度算法。

表 9-2 中比较了在 MANET 中使用编码与不使用编码时的延迟 – 吞吐量权衡。实现这些权衡的具体条件本章不作赘述，读者可以参考相应文献。

表 9-2　在 MANET 中使用编码与不使用编码时的延迟 – 吞吐量权衡

移动模型	使用编码	不使用编码
平面 – i. i. d.（快 & 单播）	$\Theta\left(\sqrt{\frac{D}{n}}\right)$ [34, 35]	$\Theta\left(\frac{n}{D}\right)$ [23]
平面 – i. i. d.（慢 & 单播）	$\Theta\left(\sqrt[3]{\frac{D}{n}}\right)$ [34]	$\Theta\left(\sqrt[3]{\frac{D}{n\log^{9/2} n}}\right)$ [21]
随机 – 步行（慢 & 单播）	当 $D=\Theta(n)$ 时，$\lambda=\Theta(1)$ [18, 35]	当 $D=\Theta(n\log n)$ 时，$\lambda=\Theta(1)$ [8]
平面 – i. i. d.（快 & 多播）	$\Theta\left(3\sqrt{\frac{D}{n_s}}\right)$ [37]	当 $D=\Theta(\sqrt{n_s p\log p})$ 时，$\lambda=\frac{1}{p\sqrt{n_s p\log p}}$ [16]

9.3.4　结论

本章总结了在不可靠无线网络中使用网络编码时的一些有趣的发现，包括在对网络规模、编码窗口大小、网络信息流数量以及应用延迟约束进行一些缩放时，不可靠无线网络的性能变化。这些结论不仅有助于对网络编码性能进行量化分析，并与传统的调度及路由策略进行比较，还有助于找到那些即使是采用简单的随机网络编码策略也能够在吞吐量、延迟或收入上获得显著增益的实例。

在本章的结尾我们要指出的是，虽然网络编码出现至今已长达十年之久，但它仍然是一个非常活跃的研究领域。我们的概述仅包含了在这个激动人心的领域中现有的和正在进行的工作的一小部分。我们相信在这个研究领域会出现更多有趣的成果，并希望本章能够对当前和未来加入该领域的研究人员带来推动性的帮助。

参考文献

[1] A. Agarwal and P. R. Kumar. Improved capacity bounds for wireless networks. Wireless Communications and Mobile Computing, 4: 251 - 261, 2004.

[2] R. Ahlswede, Ning Cai, S. R. Li, and R. W. Yeung. Network information flow. IEEE Transactions on Information Theory, 46: 1204 - 1216, July 2000.

[3] E. Ahmed, A. Eryilmaz, A. Ozdaglar, and M. M′edard. Economic aspects of network coding. In Forty - fourth Annual Allerton Conference on Communication, Control and Computing, Monticello, IL, September 2006.

[4] N. Bansal and Z. Liu. Capacity, delay and mobility in wireless ad hoc networks. In Proc. IEEE Infocom., San Francisco, CA, 2003.

[5] H. A. David. Order Statistics. Wiley Series in Probability and Statistics, 1981.

[6] S. N. Diggavi, M. Grossglauser, and D. Tse. Even one - dimensional mobility increases ad hoc wireless capacity. In Proc. IEEE Int. Symp. Information Theory (ISIT), page 352, July 2002.

[7] A. El - Gamal, J. Mammen, B. Prabhakar, and D. Shah. Throughputdelay trade - off in wireless networks. In Proc. IEEE Infocom., pages 475 - 485, 2004.

[8] A. El - Gamal, J. Mammen, B. Prabhakar, and D. Shah. Optimal throughput - delay scaling in wireless networks - part I: The fluid model. IEEE Trans. Inform. Theory, 52 (6): 2568 - 2592, June 2006.

[9] A. El - Gamal, J. Mammen, B. Prabhakar, and D. Shah. Optimal throughput - delay scaling in wireless networks - part II: Constant - size packets. IEEE Trans. Inform. Theory, 52 (11): 5111 - 5116, November 2006.

[10] A. Eryilmaz, A. Ozdaglar, and M. M′edard. On delay performance gains from network coding. In Proceedings of Conference on Information Sciences and Systems (CISS), Princeton, NJ, March 2006.

[11] A. Eryilmaz, A. Ozdaglar, M. M′edard, and E. Ahmed. On the delay and throughput gains of coding in unreliable networks. IEEE Transactions on Information Theory, 54: 5511 - 5524, December 2008.

[12] M. Ghaderi, D. F. Towsley, and J. F. Kurose. Reliability gain of network coding in lossy wireless networks. In INFOCOM, pages 2171 - 2179, 2008.

[13] M. Grossglauser and D. Tse. Mobility increases the capacity of ad hoc wireless networks. In Proc. IEEE Infocom., volume 3, pages 1360 - 1369, April 2001.

[14] P. Gupta and P. Kumar. The capacity of wireless networks. IEEE Trans. Inform. Theory, 46 (2): 388 - 404, 2000.

[15] T. Ho and D. Lun. Network Coding: An Introduction. Cambridge University Press, 2008.

[16] C. Hu, Xinbing Wang, and F. Wu. Motioncast: On the capacity and delay trade – offs. In ACM MobiHoc 09, New Orleans, May 2009.

[17] R. Koetter and M. M'edard. Beyond routing: An algebraic approach to network coding. IEEE Transactions on Information Theory, 11: 782 – 795, October 2003.

[18] Z. Kong, E. M. Yeh, and E. Soljanin. Coding improves the throughputdelay trade – off in mobile wireless networks. In Proc. IEEE Int. Symp. Information Theory (ISIT), pages 1784 – 1788, 2009.

[19] S. – Y. R. Li, R. W. Yeung, and Ning Cai. Linear network coding. IEEE Transactions on Information Theory, 49: 371 – 381, February 2003.

[20] X. Lin, G. Sharma, R. R. Mazumdar, and N. B. Shroff. Degenerate delay – capacity trade – offs in ad hoc networks with Brownian mobility. Joint Special Issue of IEEE Transactions on Information Theory and IEEE/ACM Transactions on Networking on Networking and Information Theory, 52 (6): 2777 – 2784, June 2006.

[21] X. Lin and N. Shroff. Towards achieving the maximum capacity in large mobile wireless networks. J. Commun. and Networks, 4: 352 – 361, 2004.

[22] J. Mammen and D. Shah. Throughput and delay in random wireless networks with restricted mobility. IEEE Trans. Inform. Theory, 53 (3): 1108 – 1116, March 2007.

[23] M. J. Neely and E. Modiano. Capacity and delay trade – offs for ad hoc mobile networks. IEEE Trans. Inform. Theory, 51 (6): 1917 – 1937, 2005.

[24] D. Nguyen, T. Nguyen, and B. Bose. Wireless broadcasting using network coding. In Proceeding of NetCod, 2007.

[25] S. I. Resnick. Extreme Values, Regular Variation, and Point Processes. Springer, 1987.

[26] S. M. Ross. Stochastic Processes. John Wiley & Sons, 1995.

[27] C. E. Shannon. A mathematical theory of commmunication. Bell Systems Techical Journal, 27: 379 – 423, 623 – 656, July – October 1948.

[28] G. Sharma, R. Mazumdar, and N. Shroff. Delay and capacity tradeoffs in mobile ad hoc networks: A global perspective. In Proc. IEEE Infocom., pages 1 – 12, April 2006.

[29] TerraNet. http://terranet.se/.

[30] S. B. Tirumala, A. Eryilmaz, and N. Shroff. Throughput – delay analysis of random linear network coding for wireless broadcasting. In IEEE International Symposium on Network Coding (NetCod), 2010.

[31] S. B. Tirumala, A. Eryilmaz, and N. Shroff. Throughput – delay analysis of random linear network coding for wireless broadcasting, 2011. Technical Report, submitted to IEEE Transactions on Information Theory.

[32] S. Toumpis and A. J. Goldsmith. Large wireless networks under fading, mobility, and delay constraints. In Proc. IEEE Infocom., volume 1, pages 619 – 627, 2004.

[33] W. Xiao and D. Starobinski. Extreme value fec for reliable broadcasting in wireless networks. In INFOCOM, 2009.

[34] L. Ying, S. Yang, and R. Srikant. Optimal delay – throughput tradeoffs in mobile ad hoc networks. IEEE Trans. Inform. Theory, 9 (54): 4119 – 4143, September 2008.

[35] C. Zhang, X. Zhu, and Y. Fang. On the improvement of scaling laws for large - scale MANETs with network coding. IEEE Journal on Selected Areas in Communications, 27 (5): 662 - 672, 2009.

[36] P. Zhang, C. M. Sadler, S. A. Lyon, and M. Martonosi. Hardware design experiences in zebranet. In Proc. the 2nd International Conference on Embedded Networked Sensor Systems, pages 227 - 238, 2004.

[37] S. Zhou and L. Ying. On delay constrained multicast capacity of large - scale mobile ad hoc networks. In Proc. IEEE Infocom. Mini - Conference, San Diego, CA, 2010.

第 10 章　网络编码在容断网络中的应用

Xiaolan Zhang 美国，纽约，福特汉姆大学，计算机与信息科学系

Giovanni Neglia 法国，索菲亚科技园，INRIA

Jim Kurose 美国，马萨诸塞州，马萨诸塞大学阿默斯特分校，计算机科学系

摘要

容断网络（DTN）场景通常来源于移动无线网络，这是因为移动无线网络的发送功率有限、节点移动迅速、节点密度稀疏、设备故障频繁，且从源节点到（若干）目的节点之间往往没有实时路径。本章将回顾关于在 DTN 中应用随机线性网络编码（RLC）的一些研究工作。首先要回顾的是对于 DTN 中的广播和单播应用，传统的非编码路由方案和 RLC 方案的基本操作。其次，在引入 DTN 路由的设计空间后，将集中关注在广播通信和单播通信中对基于 RLC 的路由方案的性能评估。使用基于 RLC 的方案可以改善这两种通信在能量消耗与传输性能之间的权衡关系。此外，我们还总结了关于对 RLC 方案进行性能建模、优先编码协议以及基于 RLC 的安全单播方案的研究成果。在本章的结尾，对关于网络编码在 DTN 中的应用的一些开放性问题进行了讨论。

关键词

网络编码、随机线性编码、容断网络、性能权衡、广播通信、单播通信。

10.1　引言

近年来，无线通信技术被越来越多地应用在一些没有通信基础设施的具有挑战性的环境中，在为追踪野生动物[22, 43]建设并部署的无线传感器网络、水下传感网络[38, 41]、救援队网络，为发展中国家的偏远地区或农村地区建设的网络[8, 10, 41]、车载网络[4, 20]以及手持交换网络[29]等方面的尝试证明了这一点。在没有基础设施支撑的情况下，这些网络仅依靠于无线设备之间的对等连接来支持数据通信。发送功率有限、节点移动迅速、节点密度稀疏以及设备故障频繁导致许多这类网络的连接都是间歇性的，而且节点之间会频繁地断开。容断网络（DTN，或容延网络）中通常没有从源节点到目的节点的实时路径。DTN 中的端到端通信采用的是所谓的“存储－运送－转发”范式——接收到某个数据包的节点会缓存该数据包，并在节点移动的时候搬运它，然后将该数据包传递给它遇到的新的节点。当某个携带该数据包的节点遇到该数据包的目的节点时，它就会将这个包交付出去。

由于严重的资源限制，除了间歇性连接和动态拓扑以外，在 DTN 中路由还面临着更多的挑战：对于动物或人携带的小型移动节点，其缓存空间、传输带宽以及功率都是十分有限的；对于车辆网络中的节点，尽管缓存空间及功率通常不会受到太多的限制，

但传输带宽仍然是稀缺的资源。

目前已有大量的工作研究了网络编码为无线网络带来的好处。对于有线或无线静态网络中的多播应用，Lun 等人[35]以及 Wu 等人[52]的研究表明使用网络编码可以简化最小能量多播问题（从一个 NP 完全问题简化为一个多项式时间内可解的线性优化问题）。对于移动无线网络和静态无线网络中的广播应用，Widmer 等人在参考文献［49，50］中提出了基于随机线性编码（RLC）[17,18]的高效节能广播方案。对于静态无线网络中的单播应用，一些研究工作[24,25,34,53]表明网络编码方案能够利用无线信道的广播特性来提供吞吐量增益。

考虑到 DTN 的上述鲜明特点，前文提到的网络编码为一般无线网络带来的好处在 DTN 中是不成立的。首先，DTN 的拓扑结构的动态变化使得参考文献［35，52］中采用的静态网络模型以及得出的编码为多播应用带来好处的结论均不能直接应用于 DTN。其次，DTN 中的节点密度稀疏，通常每个节点在任意时间实例内最多只有一个邻居节点。因此，前文发现的网络编码对于提升网络吞吐量（利用无线传输的广播特性）的帮助在 DTN 中是可以忽略不计的。但另一方面，DTN 中也存在着新的网络编码机会。拓扑结构的快速变化以及基础设施的缺乏都要求 DTN 中的路由方案是分布式的；此外，有限的连接及带宽又使得 DTN 中的路由方案具有局部性（即只有对局部邻域中的节点的有限的了解）。参考文献［36］中的研究表明，网络编码除了能够为提升吞吐量和节省成本带来好处以外，还有助于设计高效分布式的路由方案。

目前的关于在 DTN 中进行网络编码的工作都集中于研究将随机线性编码（RLC）应用在广播和单播通信中。下文我们用 RLC 方案表示采用了 RLC 的 DTN 路由方案，用非编码方案表示传统的路由方案。对于全部节点都需要接收所有数据包的广播应用，Widmer 等人在参考文献［49，50］中证明了 RLC 方案在提高能效方面带来的好处。对于单播传输，参考文献［55，57］证明了 RLC 方案能够更快地传播一批单播数据包。此外，将 RLC 方案与二进制喷射等待（Spray - and - wait）方案相结合来控制网络的传输次数可以改善交付延迟与开销之间的权衡[31,55,57]。最后，对于有拜占庭攻击的 DTN，参考文献［42］基于网络编码提出了能够支持安全数据通信的路由方案。

本章的其余部分结构如下，10.2 节中讲述了 DTN 的基本背景、基于非编码的广播和单播路由方案以及随机线性编码。在 10.3 节中，通过讨论 DTN 路由的各种设计角度介绍了对 DTN 路由方案的分类。10.4 节和 10.5 节中分别给出了关于网络编码为 DTN 的广播应用和单播应用带来的好处的研究结果。10.6 节讨论了关于网络编码在 DTN 中的应用的一些开放性问题。最后，10.7 节对本章进行了总结。

10.2 容断网络与随机线性编码的背景介绍

本节首先介绍所使用的网络模型，接着回顾在 DTN 中进行广播通信和单播通信时已有的基于非编码的路由方案，最后介绍 RLC 的基本操作。

10.2.1 网络模型

考虑一个由 $N+1$ 个移动节点组成的网络，网络中的节点在一个封闭的区域中独立

地移动，其移动轨迹服从某个常见的移动模型，例如随机路径点或随机方向模型[5]。每个节点都装备了传输范围相同的无线电台，当两个节点进入彼此的传输范围（相遇）时，它们能够相互交换信息。这个传输机会的持续时间被称为这两个节点的相遇时间，两次连续相遇之间的时间间隔被称为相遇间隔时间，也就是从两个节点离开彼此传输范围开始直到它们能够再次通信为止之间的这段时间。参考文献［12］中的研究表明，在随机路径点和随机方向模型下，如果相比于区域规模来说，节点速度较大而传输范围较小，则相遇间隔时间近似服从指数分布。这个指数相遇间隔时间移动模型由于其易处理性而被广泛使用（参见［31，55，57］及其中的参考文献）。

在一定时间内，DTN 中的所有的节点到节点的接触列表被称为一个 DTN 接触轨迹。我们采用由 Kempe 等人最先提出的所谓的时间网络模型[26]来表示一个 DTN 接触轨迹。该时间网络可以用一个多重图 $G=(V, E)$ 表示，其中 V 表示网络中的节点，E 表示所有边的集合，每条边都代表一次点到点的接触。给每条边 $e\in E$ 都标注一个有序对（$t(e)$，$bw(e)$），其中 $t(e)$ 指出了该边的两个端节点在什么时刻可以通信，$bw(e)$ 则指出了该次接触的带宽约束，也就是通过该次接触能够交换的数据包的数量。如果某条边的两个方向上用于传输的无线信道是相互独立的，则该边是有向的；如果某条边的两个方向上用于传输的无线信道是相同的，且总容量可以在两个方向之间任意分配，则该边是无向的。例如，图 10-1 所示的是由 4 个节点组成的 DTN 在一定时间间隔内的时间网络模型，图中的有向边表示其接触轨迹[0, 24]。

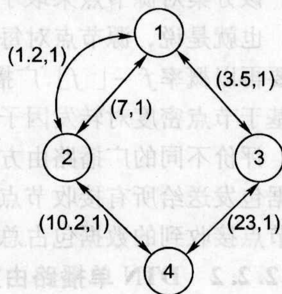


图 10-1　表示节点之间的接触轨迹的时间网络

研究人员已经对 DTN 中的广播通信和单播通信都进行了研究。在广播通信中，每个消息都需要被交付给网络中的所有节点，例如内容分发服务以及路由消息传播；而在单播通信中，每个消息都只被运送给网络中的单个节点。对基于网络编码的 DTN 路由方案的研究是从比较简单的情况开始的，即先考虑网络中只传输一代/批（广播或单播）数据包的情况。根据上述简单情况下的研究结论，在参考文献［55，57］中对一些更实际的场景进行了研究，例如数据包到达过程连续的多个单播流的场景。

10.2.2　DTN 路由方案回顾

回想一下，DTN 路由采用的是所谓的“存储 - 运送 - 转发”范式。在该范式下，网络中的每个节点都存储其他节点转发给它的数据包，当该节点四处移动时运送该数据包，并在其他中继节点或目的节点进入该节点的传输范围时将该数据包转发出去。本节中，将回顾参考文献中已经提出的广播和单播路由方案的基本操作，不同的设计选择将在 10.3 节中详细讨论。

10.2.2.1　DTN 广播路由方案

长期以来人们一直在无线 Ad Hoc 网络的背景下对广播进行研究。早期的广播路由方案是基于洪泛的，也就是网络中的每个节点对其接收到的每个消息都进行广播转发。

在相对密集的网络中，这会导致额外的开销和竞争[40]。使用基于流言机制的路由方案[13]时，每个节点都以一定的概率（即转发概率）广播转发它接收到的消息，研究表明该方案能够显著降低路由协议开销，也就是网络中发送的消息数量。

在 DTN 中，任何时刻一个节点的周围都可能没有邻居节点，因此，广播路由方案需要根据采用的控制信号（见 10.3 节）来决定什么时候对数据包进行广播转发。此外，由于每次广播转发都只能到达 0 个或 1 个邻居节点，因此节点需要对一个数据包进行多次广播转发，才能让网络中一定比例的节点接收到该数据包。出于上述原因，参考文献［49］中对流言机制进行了扩展，将转发概率推广为允许取值大于 1 的转发因子 f，并提出了概率路由。当 $f\leqslant 1$ 时，节点以概率 f 对它接收到的数据包进行一次广播转发；当 $f>1$ 时，节点对该数据包进行 $\lfloor f \rfloor$ 次广播转发，然后再以概率 $f-\lfloor f \rfloor$ 广播一次。该方案对源节点采取了特别措施，使得每个数据包在源节点上都至少会被发送一次，也就是说，源节点对每个数据包都进行 $\max(1,\lfloor f \rfloor)$ 次广播转发，且当 $f>1$ 时需要再以概率 $f-\lfloor f \rfloor$ 广播一次。由于转发因子直接决定一个方案的转发开销，因此应基于节点密度对转发因子进行调整㊀。

评价不同的广播路由方案[11, 49]的性能指标包括：（i）能量效率：通常通过将一个数据包发送给所有接收节点所需的总传输次数来衡量；（ii）数据包投递率：被所有目的节点接收到的数据包占总数据包的比例；（iii）数据包投递延迟。

10.2.2.2　DTN 单播路由方案

在单播通信中，（源节点产生的）每个数据包都会被送往网络中的单个节点（它的目的节点）。

DTN 中的单播路由方案可以分为单副本方案和多副本方案。单副本方案[46]沿单条路径转发每个数据包（不复制），任意时刻网络中都只有该数据包的一个副本。单副本方案的传输开销以及对节点缓存空间的需求都很小。然而，在预先不知道将来的相遇过程的情况下，该方案可能会作出错误的转发决定，这通常会导致其性能不够理想。对于这种情况，使用多副本方案通常可以以较大的传输开销和缓存占用为代价来减少投递延迟并提高投递率。多副本方案将数据包复制给其他节点，也就是沿多条到目的节点的路径同时对数据包进行转发；在给定的时间点，网络中通常同时存在一个数据包的多个副本。例如，Vahdat 和 Becker 提出的传染路由[48]为了投递一个数据包需要洪泛几乎整个网络。当网络的负载较小时，传染路由通过利用所有的传输机会实现最小投递延迟；但是当网络负载较大时，这会引起资源竞争。随后研究人员提出并研究了传染路由的许多变体，通过牺牲投递延迟来换取更低的资源消耗，这些变体包括 K－跳方案、机会转发方案[14]以及喷射等待方案[44, 45, 47]。

在多副本路由方案下，当数据包首次被送达目的节点时，网络中可能还存在着多个该数据包的副本。参考文献［14］中提出的回收方案可以从网络中删除这些过时的副本，

㊀ 参考文献［11］提出了一个自适应方案，该方案中节点能够基于对 2 跳之内的邻居节点的了解动态地调整其转发因子。

从而释放存储空间并避免无效的传输。10.3 节中给出了关于各种回收方案的更多细节。

在单播应用中，我们感兴趣的性能指标是数据包投递延迟以及网络中为该数据包产生的副本或组合的总数量。后者直接表征了路由方案的传输开销。显然，在数据包投递延迟和产生的副本数量之间存在一种权衡关系[56]。

10.2.3　随机线性编码

接下来将简要描述随机线性编码（RLC）的基本操作，这是在目前所有关于在 DTN 中应用网络编码的研究工作中都需要考虑的。关于 RLC 的更加正式的介绍，请参考本书的第 1 章。

假设所有数据包的长度都相同，有效载荷为 P 比特。在分组数据网络中应用 RLC 时，每个数据包的有效载荷都可以看做是大小为 q 的有限域$\mathbb{F}_q$[29]上的一个矢量，具体地说，一个 P 比特的数据包可以看做是$\mathbb{F}_q$上的一个 $d=\lceil P/\log_2(q)\rceil$维矢量。

我们称有可能被网络节点放在一起进行线性编码的一批数据包为一个分代。考虑数量为 K 的一代原始数据包 $\boldsymbol{m}_i\in\mathbb{F}_q^d$，$i=1, 2, \cdots, K$。这 K 个数据包的一个线性组合是：

$$\boldsymbol{x} = \sum_{i=1}^{K}\alpha_i\boldsymbol{m}_i, \alpha_i \in \mathbb{F}_q$$

其中的加法和乘法操作都在$\mathbb{F}_q$上进行。系数 $\alpha=(\alpha_1, \cdots, \alpha_K)$ 被称为编码向量，产生的线性组合 $\boldsymbol{x}$ 被称为编码消息。如果两个或更多的编码消息的编码向量是线性独立的，那么就说这些编码消息之间线性独立。每个原始数据包 $\boldsymbol{m}_i$ 都可以看做是一个特殊的线性组合，组合系数满足：$\forall j\neq i$，$\alpha_i=1$ 且 $\alpha_j=0$。

在基于 RLC 的路由方案下，编码系数与相应的编码消息将会一起被网络节点存储和转发。对于数量为 K 的一代数据包来说，其系数占据了 K 个符号；每个数据包或组合占据了 d 个符号，$d=\lceil P/\log_2(q)\rceil$，因此相对开销（即编码系数与数据包的大小之比）为 $K/(\lceil P/\log_2(q)\rceil)\approx K\log_2(q)/P$。如果某节点运送的编码消息的集合中最多包含 r 个线性独立的编码消息 $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_r$，我们就说这个节点的秩是 r，并称行向量是 r 个编码向量的 $r\times K$ 阶矩阵 A 为该节点的编码矩阵。从本质上来说，该节点存储的是 r 个独立的线性方程，方程的未知变量是 K 个源数据包，即 $\boldsymbol{AM}=\boldsymbol{X}$，其中 $\boldsymbol{M}=(\boldsymbol{m}_1, \cdots, \boldsymbol{m}_K)^{\mathrm{T}}$ 是由 K 个原始数据包组成的 $K\times 1$ 阶矩阵，$\boldsymbol{X}=(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_r)^{\mathrm{T}}$ 是由 r 个编码消息组成的 $r\times 1$ 阶矩阵。当一个节点（如目的节点）的秩达到 K（即满秩）时，它就可以通过矩阵求逆解码出 K 个原始数据包，也就是用标准高斯消元算法求解 $\boldsymbol{AM}=\boldsymbol{X}$，得到 $\boldsymbol{M}=\boldsymbol{A}^{-1}\boldsymbol{X}$⊖。

我们以从节点 u 到节点 v 的传输为例来说明数据的转发。用 $x_1, \cdots, x_r$ 表示节点 u 当前存储的线性组合，节点 u 从这些组合中产生一个新的随机线性组合（x_{new}），满足 $x_{\text{new}}=\sum_{j=1}^{r}\beta_j x_j$，其中系数 $\beta_1, \cdots, \beta_r$ 随机均匀地取自$\mathbb{F}_q$。显然，x_{new}也是 K 个原始数据

⊖ 如果编码矩阵 **A** 包含一个有且只有一个非零系数的矢量，则目的节点在矩阵达到满秩之前就可以解码出一个原始数据包。

包的一个线性组合。这个新的组合与其相对于原始数据包的系数一起被转发给节点 v。如果储存在节点 u 的组合中至少有一个不能被存储在节点 v 的组合线性表示，那么我们就说节点 u 具有对节点 v 有用的（即新生的）信息，且新的随机组合 x_{new} 对于节点 v 有用（可以提高节点 v 的秩）的概率大于或等于 $1-1/q$[9]。⊖

节点进行随机线性组合或者目的节点进行解码操作都会引入计算开销。编码操作的复杂度随分代大小呈线性增长，而解码操作的复杂度则随分代大小呈二次方增长。

10.3 设计空间

在给出关于 RLC 有益于 DTN 的主要研究成果之前，我们先讨论对 DTN 路由方案的不同的设计选择，以便于探索 DTN 路由方案的设计空间。除了分代管理以外的所有设计选择都可以同时应用于非编码方案和 RLC 方案。除了回收方案以外的所有设计选择都可以同时应用于广播方案和多播方案。

1. 分代管理

基于编码的方案需要解决的一个问题是：多少数据包以及哪些数据包可以构成一个分代，这就是分代管理。我们不能任意地将数据包编码到一起的原因有两个：首先，发送及存储编码系数所花费的开销随分代大小的增大而增大，解码算法的计算复杂度也是如此；其次，在单播应用中，当来自 K 个不同的（单播）流的数据包被编码到一起时，目的节点为了解码出一个需要被交付给它的数据，必须要接收 K 个编码数据包。对于每个消息都需要被交付给网络中的所有节点的广播通信，第二个考虑是不成立的。

2. 控制信令

DTN 的 Ad Hoc 特性以及动态变化的拓扑使得网络中的节点需要通过收发信标（周期性地广播信标数据包）来发现它们的邻居节点，以及/或是与邻居节点交换关于它们所运送的数据包/编码数据包的信息。节点可以根据这些控制信令来决定是否发送以及发送什么信息。本文中考虑了如下几个等级的信令控制：

- 无信令：在这个最基本的情况（在参考文献［49］中被称为无信标）下，节点没有关于邻居节点的可利用信息。节点将在不知道邻居节点是否存在的情况下决定数据包的发送。
- 常规信令：在常规信令（在参考文献［49］中被称为正常信标）下，为了发现邻居节点，也就是传输范围内的节点，每个节点都将周期性地发送信标消息。使用常规信令时，节点通常在检测到至少一个邻居节点后才发送信息。
- 全信令：在全信令（在参考文献［49］中被称为智能信标）下，节点不仅周期性地发送信标以发现邻居节点，而且还与邻居节点交换关于本地存储了哪些数

⊖ 节点 u 在已知节点 v 的编码矩阵的条件下，可以从它存储的组合中反复产生随机线性组合，直到得到一个对节点 v 有用的组合为止。或者，节点 u 也可以使用参考文献［21］中提出的确定性算法来产生对节点 v 有用的组合。这样做可以通过牺牲一部分计算开销来节省传输带宽。

据包或编码数据包的信息，也就是数据包的序列号或编码数据包的编码向量。基于这些信息，网络节点可以只将有用的信息发送给邻居节点。

3. 复制控制

对于资源受限的DTN来说，其中的节点通常是能量有限或传输带宽有限，或是二者均有限。对于这样的网络，通过所谓的复制控制机制来控制一个（或一代）数据包在网络中被发送的总次数是十分有好处的。

参考文献［49］中针对广播应用提出了一种概率路由方案，该方案通过转发因子来实现复制控制。

参考文献［45，47，55］中的喷射等待方案则采用了另一种复制控制机制，该机制可以直接控制数据包在网络中被发送的总次数。在二进制喷射等待方案[45, 47]中，源节点将为其产生的每个源数据包都分配一个计数值（一些令牌），用 L 表示，该值指定了网络中允许产生的该数据包副本的最大数量。当携带令牌值为 l（$l>1$）的数据包的节点与另一个没有携带该数据包副本的中继节点相遇时，前者会将该数据包转发给后者，且令牌值 l 将被平分给该数据包的两个副本㊀。当节点携带的数据包的令牌值为1时，该节点将不会把该数据包转发给任何中继节点，而是只把它交付给目的节点。在这种方式下，整个网络中产生的该数据包的副本的总数量以 L 为上限。当使用回收方案时，实际产生的副本数量通常会小于 L。我们将在10.5.2节中回顾并比较两个不同的复制控制方案，这两个方案的提出是为了与RLC结合使用，且均基于二进制喷射等待方案。

4. 传输调度与缓存管理

在资源受限的DTN中采用的路由方案需要通过传输调度与缓存管理[3, 28]来处理资源争用问题。当一个节点遇到另一个节点时，将由调度机制来决定前者从其缓存的所有数据包或分代中选择哪些发送给后者。当一个缓存已满的节点接收到一个新的原始数据包或编码数据包时，将由缓存管理策略来决定是否以及如何为这个新数据包挪出缓存空间。不同的传输调度与缓存管理方案将会导致不同的系统性能，比如系统范围的平均交付延迟等。

对于非编码方案，参考文献［31］中考虑了以下几种不同的调度策略：

- 随机策略——以相同的概率随机选择每个数据包；
- 本地最少策略——节点选择的发送数据包满足：到该时刻为止，该节点发送该数据包的次数最少；
- 全局最少策略——这是一个预言性方案，节点选择的发送数据包满足：网络中该数据包的副本数量最少。

参考文献［55，57］聚焦于网络编码的潜在的好处，研究了对源数据包的轮询调度策略以及对中继数据包的随机传输调度策略。也就是说，在相遇过程中，源节点依次发送每个源数据包㊁，中继节点则随机均匀地选择数据包或分代进行发送。

㊀ 如果 l 为奇数，则分配给较早副本的令牌值为 $\lceil l/2 \rceil$，分配给新副本的令牌值为 $\lfloor l/2 \rfloor$。

㊁ 当源节点采用轮询调度时，所有的源数据都有相同的机会被传播到网络中，因此得到的分块交付延迟要小于使用纯粹的随机调度的情况[57]。

对于缓存管理，参考文献［55，57］中考虑了一种活顶策略：当一个缓存已满的节点接收到一个数据包时，它会丢弃在其缓存中停留时间最长的一个中继数据包[⊖]。在RLC方案下，当缓存已满的节点接收到一个线性组合时，该节点会把它随机与缓存中已有的一个线性组合进行组合，并替换已有的线性组合。

5. 回收方案

在单播应用下，诸如传染路由和喷射等待等多副本DTN路由方案通常会使用回收方案来节省资源开销[14, 56]。例如，当使用VACCINE回收方案[14, 56]时，目的节点在首次接收到某个数据包后会产生相应的反数据包，并将该包传播到整个网络中，其目的是删除网络中失效的数据包副本，其传播方式与使用传染路由传播数据包的方式相同。在几个不同的回收方案中，VACCINE回收方案对资源的节省最为显著，因此它在现有的对网络编码在DTN单播应用中的效益的研究中被广泛地采用。具体地说，参考文献［55］在RLC方案下对VACCINE（及其他）回收方案进行了扩展：当一个分代的数据包首次被交付给它的目的节点时，目的节点会产生确认信息（被称为反分代）并将其传播到网络中，其目的是删除网络中剩余的该分代的数据包的副本或组合。

10.4　广播通信中的编码效益

在参考文献［11，49］中，作者研究了在无线网络的广播应用中使用网络编码带来的好处，并考虑了几种不同的场景，例如不同节点密度下的静态及移动无线网络。本节中给出了他们对于稀疏移动无线网络（DTN的一种）的研究成果。首先将回顾参考文献［11］中给出的理论结果，该结果表明使用RLC方案时，数据包到达所有节点所需的平均传输次数较少。接着将给出仿真研究结果，用以检验复制控制机制、控制信令等级以及移动模型对系统的影响。

10.4.1　编码有利于提高能效

相比于非编码方案，在DTN广播应用中使用基于网络编码的方案时，所有消息都被交付给所有节点所需的传输次数较少，因此后者提高了能量效率[11]。

考虑一个由N个节点组成的网络，网络中的每个节点都产生一个数据包并将它广播给其他所有节点。假设节点的移动遵循均匀随机移动模型，即每个节点在每个时隙都会独立地跳到一个均匀随机选取的新位置上。每个节点在每个时隙都分别以概率p或$p-1$来打开或关闭它的无线电台。假设不使用控制信令（没有关于邻居节点以及邻居节点所携带的消息的信息）。在每个时隙内，每个被随机打开的节点都会选择一个数据包发送（在非编码方案下），或是发送一个它所储存的编码数据包的随机线性组合（在RLC方案下）给它的邻居节点。平均每个时隙网络中会发生$(1-p)N$次传输。

定理10-1[11]　将数据包广播给所有接收者所需的平均时间：

⊖　通常假设网络节点有足够的空间来存储它们自己的原始数据包。

- 不使用网络编码时，该平均时间为 T_{w} 个时隙，$T_{w}=\frac{N\log N}{(1-p)^{2}}$；
- 使用域规模 q 足够大的网络编码时，该平均时间为 T_{nc} 个时隙，$T_{nc}=\frac{\Theta(N)}{(1-p)^{2}}$。

因此，在无编码和有编码的情况下将数据包广播给所有节点平均所需时隙的比值为

$$\frac{T_{w}}{T_{nc}}=\Theta(\log N)$$

注意，比值$\frac{T_{w}}{T_{nc}}$与在这两种情况（使用和不使用 RLC）下将数据包广播给所有接收者所需的总传输次数的比值相等，这是因为在两种方案下平均每个时隙的传输次数相等，均为 $N(1-p)$。

正如参考文献［11］中观察到的，RLC 为上述场景带来的效益与基于 RLC 的消息传播协议——代数流言协议[9]本质上是相似的。它们都可以看做是赠券收集问题的特例，也就是从 N 张不同的赠券组成的集合中有放回地抽取赠券，并研究收集全部 N 种赠券所需的抽取次数。实际上，收集所有种类的赠券平均需要抽取 $O(N\log N)$ 次[39]。随着已抽取到的赠券越来越多，下一次抽到新赠券的概率通常会越来越小。与之相比，域规模足够大的 RLC 方案可以看做是一个改版的赠券收集问题，该问题中每次抽取都能获得一张新赠券的概率较高，因此要收集全部 N 种赠券仅需 $O(n)$ 次抽取[9]。

10.4.2　实际的 RLC 广播方案

上一节考虑了均匀随机移动模型，在不使用控制信令与重复控制机制的假设下对几种简单路由方案的性能进行了比较。对于更加实际的情况，例如网络中的节点的移动遵循随机路径点移动模型，在采用控制信令以及重复控制机制的假设下，参考文献［14，49］进行了相应的仿真研究以评估 RLC 方案带来的效益。

均匀随机移动模型在每个时隙都会对所有节点重新洗牌。当节点的传输范围相比于节点的移动区域和/或节点在一个信标周期内移动的距离来说较小时（参见 10.2.1 节），诸如随机路径点模型（RWP）和随机方向模型等一般的移动模型都会表现出相似的无记忆性。我们期望 RLC 方案可以带来与均匀随机移动模型下相似的效益。但是，随着节点移动变慢以及传输范围变大，数据包将越来越不能良好地“混合”，这使得 RLC 方案的效益不再那么显著。图 10-2 中画出了在 3 种不同的移动模型下，非编码方案和 RLC

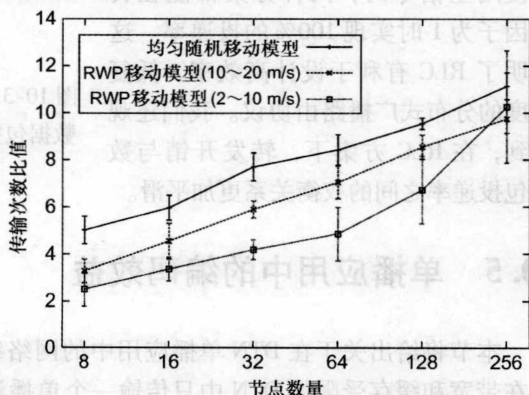


图 10-2　在均匀随机移动模型以及几个平均速度不同的随机路径点（RWP）移动模型下，使用简单的洪泛方案和 RLC 广播方案所需的传输次数的比值（见参考文献［11］中的图 10-8）

方案所需的传输次数的比值。我们观察到，RLC 方案在 RWP 下的相对增益比在均匀随机移动模型下要小。这是因为节点移动速度越慢，对节点的洗牌越少，导致 RLC 增益越小。

下面考虑当引入控制信令和复制控制机制时，使用 RLC 带来的效益。图 10-3 中所示的是当我们通过转发因子来控制复制数量时（正如 10.2.2.1 节中介绍的），在（i）概率路由（即非编码）方案和（ii）RLC 方案下实现的数据包交付率。我们观察到，在静态场景下（即标注“静态网络”的曲线），非编码方案和 RLC 方案都表现不佳。与此相反，当节点遵循暂停时间为零、最小速度和最大速度分别为 2m/s 和 10m/s 的 RWP 移动模型时，在三种控制信令下 RLC 方案均优于非编码方案。当不使用信令或仅使用常规信令时，网络编码的效益是十分显著的。例如，在不使用信令的情况下，当转发因子达到 4 时，RLC 方案能够交付 80% 的数据包，而使用非编码方案时几乎所有数据包都会交付失败。当使用全信令时，两种方案都能在转发因子为 1 时实现 100% 的投递率。这表明了 RLC 有利于设计高效率、低复杂度的分布式广播路由协议。我们还观察到，在 RLC 方案下，转发开销与数据包投递率之间的权衡关系更加平滑。

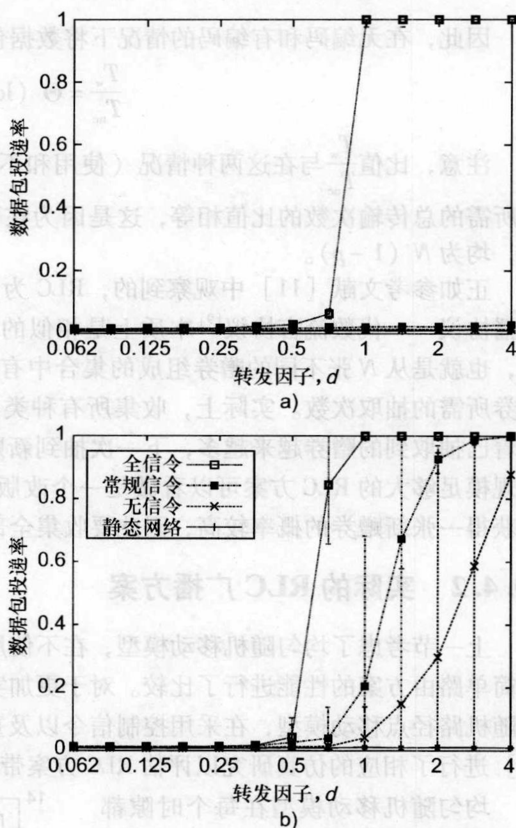


图 10-3　在稀疏网络中，转发因子取不同值时的数据包投递率（见参考文献［49］中的图 10-4）
a）概率路由方案　b）RLC 方案

10.5　单播应用中的编码效益

本节将给出关于在 DTN 单播应用中的网络编码效益的研究成果。我们主要关注的是在带宽和缓存受限的 DTN 中只传输一个单播流的简单情况，该单播流由数量为 K 的一组数据包组成。使用 4 元组（s，d，t_0，K）来表示由源节点 s 在 t_0 时刻产生的、被运送往同一个目的节点 d 的一组数量为 K 的单播数据包。假设在每次接触过程中节点都可以交换 b（$b<K$）个数据包。注意，如果没有这个限制，这组数据包就可以作为单个消息被传播，那么就没有使用网络编码的必要。假设每个节点都有足够的缓存空间来存

储它作为源发出的数据包或者以它为目的节点的数据包，但仅能够携带 B（$B<K$）个中继数据包。我们感兴趣的性能指标是分组投递延迟，也就是投递该组数据包所需的时间，以及产生的副本或组合的总数量。

10.5.1 节将说明对这组数据包应用 RLC 可以减少它的分组投递延迟。接下来在 10.5.2 节中，将证明当使用重复控制方案时，RLC 方案可以改善投递延迟与传输次数之间的权衡关系。对于多个分代的情况，它还可以减小网络范围的平均分组投递延迟。在 10.5.3 节中将讨论带宽与缓存约束、分组大小、分代成形以及不同的控制信令等级是如何影响 RLC 方案的相对效益和开销的。在 10.5.4 节中将研究对 RLC 方案的性能建模。最后，在 10.5.5 节中将回顾其他一些关于在 DTN 单播通信中应用网络编码的研究工作。

10.5.1 网络编码减少分组投递延迟

本节中将证明 RLC 方案有益于减少分组投递延迟。对于一组给定的数据包（s，d，t_0，K）以及一个固定的接触轨迹，存在一个最小分组投递延迟，该延迟可以通过使用需要所有接触信息的集中式预言性方案来实现，此外还存在一个所有路由方案都可以实现的分组投递延迟的下界（见 10.5.1.1 节）。RLC 方案中的数据转发的更高的随机性使得它能够以更高的概率实现最小分组投递延迟（见 10.5.1.2 节）。最后，在 10.5.1.3 节中讨论了 RLC 方案在其他指标上的性能。

10.5.1.1 最小分组投递延迟

对于由 4 元组（s，d，t_0，K）表示的一组数据包，即由源节点 s 在 t_0 时刻产生的被运送往目的节点 d 的一组数量为 K 的数据包，在一个给定的接触轨迹下，可以使用参考文献［57］中提出的算法来计算它的最小分组投递延迟。

如下所述，该算法根据节点之间的接触轨迹构造了一个时间独立的事件驱动图[16]（图 10-4 中所示的是与图 10-1 中的接触轨迹相对应的事件驱动图）。从一个空白的事件驱动图开始，按照时间顺序来处理接触轨迹中的每个接触事件。对于移动节点 i 和移动节点 j 在时刻 t 发生的一次接触事件，用（i，t）和（j，t）来标记这两个节点，并在图中插入一条连接它们的链路。链路上的标注表示在这次接触过程中每个方向上能够传输的数据包的数量。此外，还需要分别将两个节点（（i，t）和（j，t））连接到各自在前一次接触事件中所接触的节点上（如果存在的话）。这些边上的标注表示能够“传输”的数据包的最大数量，第一个值对应该节点的缓存大小，第二个值通常等于 0[㊀]。例如，对于节点 1 和节点 3 在 $t=3.5$ 时刻的接触事件，我们先将节点（1，3.5）和（3，3.5）插入图中，再用标注了（1，1）的链路将它们连接起来。此外还需将节点（1，3.5）连接到节点（1，1.2）上，假设每个节点都能够存储 B 个中继数据包，则连接链路的标记为（B，0）。为了计算一组用 4 元组（s，d，t_0，K）表示的数据包的分组投递延迟，

㊀ 节点 i 在时刻 t 接收到的数据包不可能被传输给过去时刻的该节点。

需要在图中添加两个特殊的节点（s，t_0）和 d（例如图10-4中的节点（1，0）和节点4）。

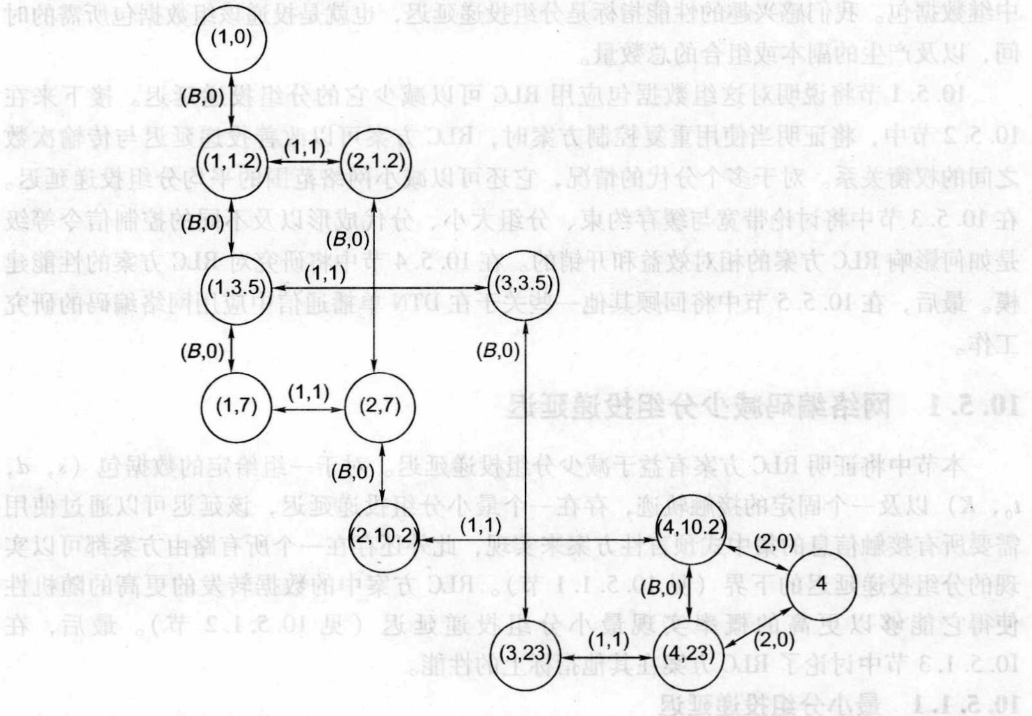


图10-4　在图10-1所示的接触轨迹下，用于计算分组（1，4，0，2）的最小分组投递延迟的事件驱动图。假设每个节点最多可以存储B个中继数据包。图中以粗线画出的是用于传输随机线性组合的链路（有 $\eta=4$ 条这样的链路）。例如，节点1通过链路（1，1.2）→（2，1.2）发送一个由两个数据包组成的随机线性组合给节点2。节点3在链路（3，23）→（4，23）上的传输不使用RLC，这是因为它的缓存中只有一个编码数据包

事件驱动图是一种静态图，它不仅可以表示各个接触事件在接触轨迹中的时间顺序，还可以表示带宽和缓存限制。在某个接触轨迹下的数据包传播与其静态图中的数据包传播是相对应的：一个移动节点（用节点1表示）在时间点（$t=1.2$）发送一个（编码或原始）数据包给另一个移动节点（用节点2表示）可以表示为该数据包沿水平链路（即（1，1.2）→（2，1.2））的传输；节点2缓存一个数据包直到在时间点（$t=7$）发生另一次接触事件，这在静态图中对应着该数据包沿垂直链路（即（2，1.2）→（2，7））的传输。我们可以对该静态图采用经典的图论算法[16]来解决各种网络问题。参考文献［57］中说明了如何通过事件驱动图来计算（s，d，t_0，K）的最小分组投递延迟。首先我们观察到，如果在事件驱动图中存在一条从（s，t_0）到 d 的值为 k 的流，那么就存在一组能够从源节点传输 k 个数据包给目的节点的转发决策，该转发决策利用的是用于生成事件驱动图的接触事件。具体地说，如果这些接触发生在时间范围［t_0，t］内，那么存在值为 k 的流就意味着有可能可以在 t 时间之前传输 k 个产生

于时间点 t_0 的数据包。鉴于上述考虑，(s, d, t_0, K) 的最小分组投递时间 t' 是满足以下条件的最早的 t'：根据在时间范围 $[t_0, t']$ 内发生的接触事件可以建立一个事件驱动图，该图中存在一条从 (s, t_0) 到 d 的值为 K 的流。最小分组投递延迟等于最小分组投递时间 t' 减去开始时刻 t_0 所得的值。

参考文献［57］中提出的算法能够迭代地找到最小分组投递延迟。该算法从一个空的事件驱动图开始，扫描分组产生时间 t_0 之后发生的接触轨迹，按照时间顺序处理各个接触事件并逐渐扩大该事件驱动图，直到在该图中找到一组从节点 (s, t_0) 到节点 d 的总容量为 K 的路径（使用 Ford - Fulkerson 算法[7]）。上述过程结束后，该算法最后处理的接触事件的时间就是最小分组投递时间。

10.5.1.2　实现最小分组投递延迟的概率

下面研究随机非编码和编码方案在实际场景中的表现，场景中假设节点没有掌握将要发生的接触事件的信息。仍然考虑图 10-1 所示的接触轨迹，以一组数据包（1，4，0，2）为例进行说明。该组数据包的最小分组投递延迟是 23，因为只有在处理完发生在 $t=23$ 时刻的接触事件后，事件驱动图（见图 10-4）中从节点（1，0）到节点 4 的最大流才达到 2。

首先考虑非编码方案。假设中继节点使用随机调度策略，源节点使用轮询调度策略，该方案采用全信令。节点 1 首先在时间点 $t=1.2$ 转发数据包 m_1 给节点 2，接着在时间点 $t=3.5$ 转发 m_2 给节点 3㊀，最后在时间点 $t=7$ 转发 m_2 给节点 2（此时节点 2 已经有 m_1）。当节点 2 和节点 4 在时间点 $t=10.2$ 相遇时，节点 2 从它的缓存中随机选取一个数据包（m_1 或 m_2）交付给节点 4。此时可能会有两种情况：

（i）节点 2 以 0.5 的概率将数据包 m_1 传输给节点 4，当节点 3 在时间点 $t=23$ 将 m_2 交付给节点 4 时，最小分组投递延迟得到实现。

（ii）节点 2 以 0.5 的概率将数据包 m_2 传输给节点 4，当节点 3 和节点 4 在时间点 $t=23$ 相遇时，节点 3 没有对节点 4 有用的信息。因此，非编码方案实现最小延迟的概率是 0.5。

使用 RLC 方案时，节点 1 在时间点 $t=1.2$ 和 $t=7$ 分别将随机线性组合 c_1 和 c_2 发送给节点 2，并在时间点 $t=3.5$ 将 c_3 发送给节点 3。节点 3 在时间点 $t=23$ 将 c_3 发送给节点 4。依据 c_1 与 c_2 是否相互独立，分两种情况进行讨论。

（i）当 c_1 与 c_2 相互独立时，节点 2 将存储这 2 个组合，并产生一个 c_1 和 c_2 的随机线性组合 c_{12}。在与节点 4 相遇时，节点 2 会把它转发给节点 4。由于 c_3 可以用 c_1 和 c_2 线性表示（假定 c_1 和 c_2 相互独立），因此 c_3 与 c_1 和 c_2 的随机线性组合 c_{12} 相互独立的概率是 $1-1/q$。我们认为在这种情况下，节点 4 在时间点 $t=23$ 能够从 c_{12} 和 c_3 中解码出 2 个原始数据包的概率是 $1-1/q$。

（ii）当组合 c_1 与 c_2 线性相关时，节点 2 仅存储 c_1，并在时间点 $t=10.2$ 将 c_1 转发给节点 4。c_3 和 c_1 相互独立的概率也就是节点 4 在时间点 $t=23$ 达到满秩的概率为 $1-1/q$。由于这两种情况是互斥且完备的，因此对于该接触轨迹，我们认为 RLC 方案能够以概

㊀ 译者注：原文中为节点 2，译者疑有误。

率 $1-1/q$ 实现对分组（1，4，0，2）的传输的最小分组投递延迟。当有限域大小 q 取一个常用值 2^8 时，这个概率要比非编码方案实现的 0.5 的概率大得多。

从这个例子中可以看到，相比于非编码方案，RLC 方案提供了更大的随机性。这是因为网络节点随机且独立地组合它们存储的编码数据包来产生新的编码数据包进行转发，而独立的编码数据包的数量远远大于非编码方案的从其中选择一个进行发送的数据包的数量 K。其结果就是，RLC 方案转发冗余信息的概率远远小于非编码方案。

一般来说，RLC 方案实现最小分组投递延迟的概率取决于接触轨迹，更具体地说是取决于在该分组产生之后的时间内网络中发生的接触事件的序列。为了更深入地了解 RLC 方案，我们从另一个角度来看待信息的传播。对于图 10-1 所示的接触轨迹，再次考虑分组（1，4，0，2）。基于 RLC 的 DTN 路由方案与图 10-4 所示的静态图上的基于 RLC 的传输方案相对应。沿水平链路的一次传输，例如链路（2，10.2）→（4，10.2），对应的是从一个移动节点（节点 2）到另一个移动节点（节点 4）在时间点（$t=10.2$）发生的传输。如果该传输的发送节点在发送给接收节点之前对它存储的数据包进行了组合，则该传输涉及 RLC 操作。另一方面，在静态图中沿垂直链路的传输表示的是数据包被存储在移动节点中，因此该传输不涉及 RLC 操作。对于该 DTN 接触轨迹，当且仅当 RLC 方案以最小分组投递延迟交付该组数量为 $K=2$ 的数据包时，RLC 方案才能实现从节点（1，0）到节点 4 的最大流 $K=2$。前一个问题可以看做是参考文献［18］中研究的一般的多源多播连接问题的一种特殊情况。该问题考虑的是对一组多播数据包的传输，这组数据包源自不同的源节点，去往（静态）图 $G=<V, E>$ 中的一组共同的接收节点。图中的每条链路都有一个确定的容量限制。如果网络可以支持由给定的一组源节点、一组共同的接收节点以及相应的源节点数据速率构成的一个多播流，则参考文献［18］中的如下定理给出了 RLC 方案，能够支持该多播流的概率的下限。

定理 10-2[18]　考虑任意网络中的一个多播连接问题，网络中的源节点相互独立或线性相关，所采用的网络码中的一些或全部网络编码系数均匀随机地取自有限域 $\mathbb{F}_q$，其中 $q>d$（d 是多播接收者的数量），其余的编码系数，如果存在的话，是固定的。如果该网络连接问题存在一个与固定编码系数的值相同的解，那么该随机网络码对该问题有效的概率至少为 $(1-d/q)^{\eta}$，其中 η 表示与随机系数相关的链路的数量。

注意这里的 η 指的是执行 RLC 操作的源 - 目的节点路径集（总容量为 K）包含的链路的数量㊀。通过将 d 的值设定为 1，可以把上述定理应用于单播通信。假设该静态图在最小分组投递时间内可以支持一个容量等于该分组大小的流，那么基于上述定理，可以得出结论：RLC 方案实现最大流的概率的下限值为 $(1-1/q)^{\eta}$。也就是说，RLC 方案能够以大于等于 $(1-1/q)^{\eta}$ 的概率实现最小分组投递延迟。这个结果表明，对于两个看似不同的网络场景，也就是传统的静态网络以及网络拓扑动态变化的 DTN，使用 RLC 带来的效益之间存在着一些根本性的关联。

仿真结果证实，RLC 方案下的分组投递延迟与最小分组投递延迟十分接近，如图 10-5a 所示。图中分别画出了最小分组投递延迟以及 RLC 方案和非编码方案下实现的分组投递延迟的经验累积分布函数（CDF），该结果通过运行 100 次不同随机种子的仿真得到。

㊀ 当某条链路能够传输的数据包的个数 $B>1$ 时，我们就将它看做是 B 条链路。

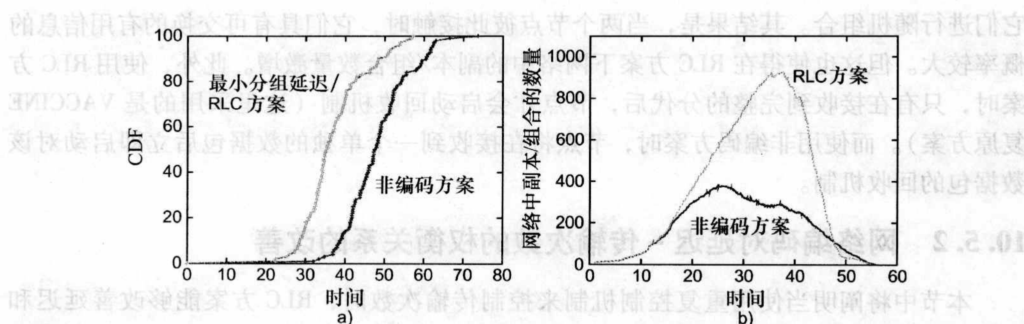


图 10-5 由 $N=101$ 个节点组成的 DTN，相遇间隔时间服从参数为 $\beta=0.0049$ 的齐次指数分布，每次接触的带宽约束为 $b=1$ 个数据包，缓存空间不限

a）分组投递延迟的 CDF，运行了 100 次仿真 b）1 次仿真中的存储消耗

10.5.1.3 其他指标

我们已经看到，相比于非编码方案，RLC 方案能够实现更小的分组投递延迟，下面考虑其他的性能指标。

首先考虑平均包延迟和平均顺序包延迟。图 10-6 所示的是在 RLC 方案和非编码方案下，运行 100 次不同的仿真得到的各个延迟指标的经验累积分布函数。我们观察到，在 RLC 方案下，分组投递延迟、平均包延迟和顺序包延迟这三个指标几乎完全相同，且均十分接近于最小分组投递延迟。另一方面，在非编码方案下，平均包延迟、顺序包延迟和分组投递延迟之间有着显著的区别。RLC 方案在分组投递延迟和顺序包延迟方面比非编码方案的表现要好（分组投递延迟性能的提升大于顺序包延迟性能的提升），但是在平均包延迟方面，RLC 比非编码的表现要差一些。注意这里考虑的 RLC 方案在具体的实施上采取了一些特殊的措施来提高它的平均投递延迟性能。例如，如果一个节点在接收矩阵满秩之前就能够解码一个或多个数据包，那么它就会把已经解码的数据包（而不是编码数据包的随机线性组合）转发给目的节点。

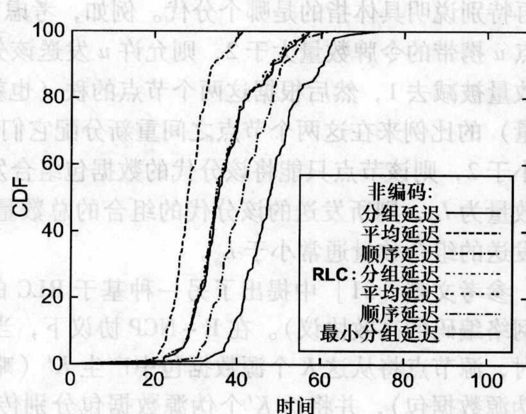


图 10-6 在与图 10-5 相同的网络设定下，运行 100 次仿真得到的各个延迟指标的累积分布函数

RLC 方案以更多的传输以及更大的缓存占用为代价换取了更快的信息传播。例如，图 10-5b 中所示的是网络中的数据包副本（非编码方案）或组合（RLC 方案）的总数量作为一次仿真的运行时间的函数。在 RLC 方案下，网络节点在转发数据包之前会对

它们进行随机组合。其结果是，当两个节点彼此接触时，它们具有可交换的有用信息的概率较大。但这也使得在 RLC 方案下网络中的副本/组合数量激增。此外，使用 RLC 方案时，只有在接收到完整的分代后，节点才会启动回收机制（这里采用的是 VACCINE 复原方案）。而使用非编码方案时，节点将在接收到一个单独的数据包后立即启动对该数据包的回收机制。

10.5.2　网络编码对延迟 – 传输次数的权衡关系的改善

本节中将阐明当使用重复控制机制来控制传输次数时，RLC 方案能够改善延迟和传输次数之间的权衡关系。也就是说，当网络中的传输次数相近时，RLC 方案能够实现比非编码方案更小的平均分组投递延迟[55, 57]。

首先考虑参考文献［55，57］中提出的基于二进制喷射等待方案的重复控制机制，也就是所谓的基于令牌的 RLC 方案。在该方案中，源节点为它产生的每代数据包都分配若干数量（用 L_g 表示）的令牌，该令牌被用于限制网络中允许交换的该分代的组合的总数量。数据的转发遵循 10.2.3 节中描述的 RLC 方案的基本操作并采用一个具体的令牌管理策略。为了简单起见，先考虑一个分代，下文讨论的令牌数量以及节点的秩都不再特别说明具体指的是哪个分代。例如，考虑节点 u 和节点 v 之间的一次接触。如果节点 u 携带的令牌数量大于 2，则允许 u 发送该分代的一个组合。发送之后节点 u 的令牌数量被减去 1，然后根据这两个节点的秩（也就是节点所携带的线性独立的数据包的数量）的比例来在这两个节点之间重新分配它们的总令牌数㊀。如果节点携带的令牌数量小于 2，则该节点只能将该分代的数据包组合发送给目的节点。假设该分代的初始令牌数量为 L_g，则所发送的该分代的组合的总数量的上限为 L_g。当使用回收方案时，实际发送的组合数量通常小于 L_g。

参考文献［31］中提出了另一种基于 RLC 的重复控制方案，被称为 E – NCP（基于网络编码的高效协议）。在 E – NCP 协议下，当源节点要发送 K 个源数据包给目的节点时，源节点将从这 K 个源数据包中产生 K'（略大于 K）个随机线性组合（我们称之为伪源数据包），并将这 K' 个伪源数据包分别传播给前 K' 个与它相遇的中继节点。这 K'㊁个中继节点将采用二进制喷射等待机制来限制伪源数据包的总传输次数。不同的伪源数据包可以在中继节点上被组合（详见参考文献［31］）。

参考文献［57］中的仿真研究比较了在非编码方案（以 K 个数据包为单位应用二进制喷射等待机制）、基于令牌的 RLC 方案以及 E – NCP 方案下实现的分组投递延迟和传输次数之间的权衡关系。图 10-7 所示的是在不同的令牌限制下，对于没有缓存限制（见图 10-7a）和满足缓存限制 $B = 2$（见图 10-7b）的两种情况，数量为 $K = 10$ 的一组

㊀ 对于两个没有信息需要交换的节点，其令牌数量也会按照秩的比例被重新分配。目的是维持节点传播分代信息的能力与它关于该分代的秩成正比这一基本原则。

㊁ 传播 K' 个伪源数据包是为了提高能够从接收到的 K 个编码包中解码出 K 个原始数据包的概率。正如参考文献［57］中指出的，对于源节点来说，传播 K 个原始数据包或 K 个线性独立的编码数据包给 K 个中继节点就足够了。

数据包的平均分组投递延迟与传输次数之间的关系。我们观察到，当传输次数相近时，RLC 方案能够实现比非编码方案更小的分组投递延迟；基于令牌的 RLC 方案的性能优于 E-NCP 方案，尤其是在传输次数较少时。对于中继节点有缓存限制时的研究结果进一步证明了 RLC 方案能够在不增加传输开销的条件下减小分组投递延迟。

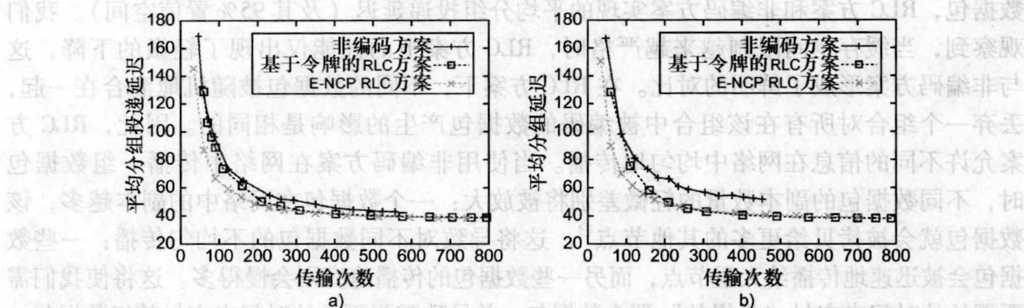


图 10-7　分组投递延迟和传输次数之间的权衡关系，除带宽限制和缓存限制外，其余的网络设定都与图 10-5 相同

a）延迟和传输次数之间的权衡关系（$b=1$，$B=\infty$）　b）延迟和传输次数之间的权衡关系（$b=1$，$B=2$）

RLC 方案可以改善投递延迟与传输次数之间的权衡关系，这一事实也解释了在多个分代的情况下观察到的 RLC 方案的效益[55, 57]。考虑网络中有多个连续的单播流的场景，每个源节点都依据泊松（Poission）过程独立地产生数据包分组，并对同一分组的数据包使用 RLC，也可以将每个分组都理解为一个分代。不论是否使用网络编码，重复控制机制都可以通过限制传输次数来减少带宽争用甚至减少系统范围内的平均分组投递延迟。令牌的数量存在一个可以使平均分组投递延迟最小的最优值：令牌数量过多或者不进行重复控制都会导致网络中出现严重的争用并降低网络性能，但是令牌数量过少又会使得网络中的节点无法充分利用所有可用的带宽。参考文献［55，57］中的研究表明，在各种通信速率下，尤其是在缓存空间受到限制时，RLC 方案实现的最小投递延迟要小于非编码方案。

10.5.3　关于 RLC 效益的讨论

本节将概述参考文献［55，57］中的仿真研究，并说明不同的系统参数对 RLC 效益的影响。

10.5.3.1　不同带宽和缓存限制的影响

我们已经看到，对于带宽和缓存受限的 DTN 网络，RLC 方案能够实现比非编码方案更小的分组投递延迟。如果逐渐放松对网络带宽的限制，RLC 的效益将呈现递减的趋势。当在所有的接触过程中能够交换的数据包的数量 b 均等于分组大小 K 时，RLC 的效益将消失。在这种情况下，K 个数据包能够独立地传播而不必竞争带宽。例如㊀，在一个由 101 个节点组成的网络中传播来自同一个单播流的 $k=10$ 个数据包，引自参考文献［57］的图 10-8a 中所示的是该情况下对于不同的带宽限制的平均分组投递延迟及其

㊀ 因此，正如参考文献［56］中阐述的，在没有资源约束的条件下，分组投递延迟将与传染路由延迟相同。

95%置信区间（运行50次不同的仿真得到）。同样地，当放松带宽限制时，RLC方案在延迟－传输权衡方面相比于非编码方案的效益也将减少。

对于中继节点的带宽和缓存空间同时受限的情况，使用RLC方案十分有益。图10-8b中所示的是在不同的节点缓存空间大小B（$B \leqslant K$）下，对于数量为$K=10$的一组数据包，RLC方案和非编码方案实现的平均分组投递延迟（及其95%置信空间）。我们观察到，当缓存空间限制越来越严格时，RLC方案下的性能仅出现了轻微的下降，这与非编码方案形成了鲜明的对比。在RLC方案下，不同的数据包被随机地混合在一起，丢弃一个组合对所有在该组合中被编码的数据包产生的影响是相同的。因此，RLC方案允许不同的信息在网络中均匀地传播。当使用非编码方案在网络中传播一组数据包时，不同数据包的副本数量的轻微差别将被放大：一个数据包在网络中的副本越多，该数据包就会被拷贝给更多的其他节点[⊖]。这将导致对不同数据包的不均匀传播：一些数据包会被迅速地传播给大量节点，而另一些数据包的传播速度则会慢得多。这将使我们需要更长的时间来交付“最慢的”那个数据包，并导致需要更长的时间来交付整组数据包。

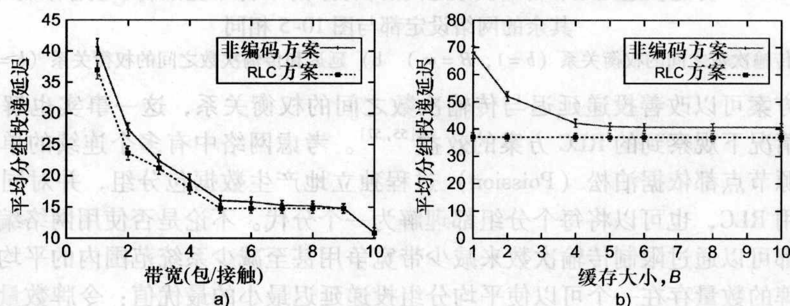


图10-8　带宽和缓存限制的影响，除带宽限制（见图a）和缓存限制（见图b）外，其余的网络设定都与图10-5相同

a）不同带宽限制下的E［Dblock］　b）不同缓存限制下的E［Dblock］

10.5.3.2　分代管理的影响

分代管理用于决定多少以及哪些数据包可以形成一个进行随机线性编码的分代，下面讨论分代管理的影响。

迄今为止，我们一直侧重于研究同一分代的数据包来自同一单播流的情况。参考文献［55，57］中探讨了其他的一些分代形成方式，包括：形成同一个分代的数据包来自多个不同的源节点但去往相同的目的节点，或者来自多个不同的源节点并去往不同的目的节点。我们分别用SS_ SD、MS_ SD和MS_ MD（单源多目的、多源单目的、多源多目的）来指代这3种情况。对MS_ SD使用RLC方案所获得的效益要小于SS_ SD。这是因为对于从K个不同的源节点开始传播K个数据包的MS_ SD，使用非编码方案时，

⊖　在包含N个节点的网络中，副本数量为n的数据包的传播速度的一级近似值正比于$n\ (N-n)$，当$n<N/2$时则有：数据包的副本数量越多，其传播速度越快。

中继节点转发数据包错误所造成的影响变得不那么显著。对于 MS_ MD 的情况，RLC 方案的表现不如非编码方案，这是因为前者将去往不同目的节点的数据包混合在了一起，这导致目的节点想要解码一个发送给自己的数据包，必须接收 K 个独立的组合。一般情况下，应限制 RLC 方案只对去往相同目的节点的数据包进行混合。

参考文献［55，57］中的研究还表明，RLC 的相对效益随分代大小 K 的增大而减小。这是因为对于非编码方案，分组大小越大，则对数据包的随机选择空间越大，因此在两条路径上选择相同的数据包进行转发的概率越小。

10.5.3.3　控制信令的影响

以上所有对单播应用的研究结果都是基于全信令的情况，也就是说，两个相遇的节点会首先交换关于它们各自携带了哪些数据包（或编码数据包）的信息，以免给对方发送无用的信息。然而，交换这种控制信息需要消耗传输带宽，尤其是对于有更多控制信息需要交换的 RLC 方案（除数据包 ID 外还需交换所有编码数据包的编码系数）。此外，对于 RLC 方案，全信令还将引入计算复杂度。这是因为每个节点都需要基于接收到的编码系数进行一些计算，以确定自己是否有对另一个节点有用的信息。

下面考虑使用常规信令的情况，也就是节点不交换关于它们所携带的数据包或编码数据包的信息。在非编码方案下，没有这些信息的节点将从它携带的数据包集合中随机选择一个转发给另一个节点；在 RLC 方案下，节点将持续产生并发送随机线性组合给另一个节点，直至另一个节点满秩。图 10-9 中分别画出了在全信令和常规信令下，使用非编码方案和 RLC 方案实现的分组投递延迟与传输次数之间的权衡关系。我们观察到，使用常规信令和使用全信令时，RLC 方案的性能几乎完全相同，而对于非编码方案，使用常规信令时的性能明显不如全信令的情况。这证明了即使在信令开销更少的情况下，即当 RLC 方案采用常规信令而非编码方案采用全信令时，RLC 方案仍然能保持其相对于非编码方案的效益。

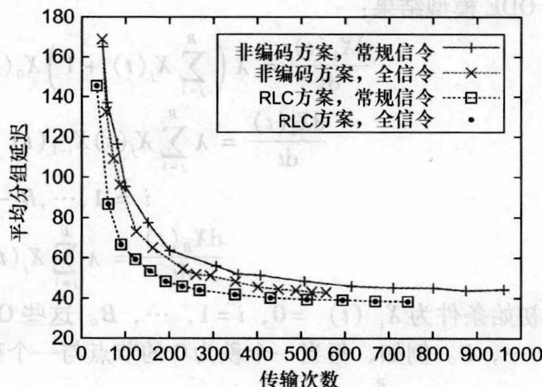


图 10-9　使用全信令和常规信令时，分组投递延迟与传输次数之间的权衡关系、网络设定与图 10-5 相同

10.5.4　对网络编码方案的建模研究

为了对 RLC 方案的效益进行定量分析，参考文献［32，33］中基于常微分方程（ODE）提出了描述非编码方案以及 RLC 方案下的分组投递延迟的模型。

该模型描述了在一个包含 $N+1$ 个移动节点的网络中，在 $t=0$ 时刻产生的一组数量为 K 的数据包从源节点向同一个目的节点的传播过程。每对节点的接触时间都遵循参

数为 λ 的指数分布。在每次接触中，2 个节点之间的两个方向上各允许传输 1 个数据包。所有节点的缓存空间都是有限的，最多可以存储 B（$B<K$）个中继数据包。关于在整个网络中传播多个流言或信息的流言算法的问题，大多数研究工作都致力于描述停止时间，也就是采用不同的流言算法时所有节点要接收所有消息所需的时间（参见文献 [37] 及其中的参考文献）。

在后文给出的一些简单假设下，对于 RLC 方案和非编码方案，系统在 t 时刻的状态都可以通过一个 B 元组 $\{X_1(t), X_2(t), \cdots, X_B(t)\}$ 来表示，其中 $X_i(t)$ 是网络中秩为 i 的节点的数量（包括源节点在内），则秩为 0 的节点的数量为 $X_0(t)=N-\sum_{j=1}^{B}X_j(t)$。

假设采用 RLC 方案以及常规信令㊀，并进一步假设：当节点接收到来自另一个节点的一个随机线性组合时，它的秩通常会增加 1，即该组合包含了对该节点有用的信息。这是一个乐观的假设，因为另一个节点可能没有对该节点有用的信息，而且即使另一个节点有对该节点有用的信息㊁，它也可能产生不包含有用信息的随机线性组合。给出以下 ODE 模型结果：

$$\frac{\mathrm{d}X_1(t)}{\mathrm{d}t}=\lambda\left(\sum_{j=1}^{B}X_j(t)+1\right)X_0(t)-\lambda\sum_{j=1}^{B}X_j(t)X_1(t) \tag{10-1}$$

$$\frac{\mathrm{d}X_i(t)}{\mathrm{d}t}=\lambda\sum_{j=1}^{B}X_j(t)X_{i-1}(t)-\lambda\sum_{j=1}^{B}X_j(t)X_i(t)$$
$$i=1,\cdots,B-1 \tag{10-2}$$

$$\frac{\mathrm{d}X_B(t)}{\mathrm{d}t}=\lambda\sum_{j=1}^{B}X_j(t)X_{B-1}(t) \tag{10-3}$$

其初始条件为 $X_i(t)=0$，$i=1, \cdots, B$。这些 ODE 方程描述了 $X_i(t)$ 的变化率，$i=1,\cdots,B$。例如，每当一个秩为 0 的节点与一个秩大于 0 的节点（包括源节点在内这样的节点总共有 $\sum_{j=1}^{B}X_j(t)+1$ 个）相遇，$X_1(t)$ 就会增加 1，因此 $X_1(t)$ 的增长速度为 $\lambda\left(\sum_{j=1}^{B}X_j(t)+1\right)X_0(t)$。另一方面，每当一个秩为 1 的节点与一个秩大于 0 的节点（包括源节点在内但不包括它自身，总共有 $\sum_{j=1}^{B}X_j(t)$ 个这样的节点）相遇，它的秩将增加 1，从而 $X_1(t)$ 将减少 1，因此 $X_1(t)$ 的下降速度为 $\lambda\sum_{j=1}^{B}X_j(t)X_1(t)$。同时考虑 $X_1(t)$ 的增长速度和下降速度可以推导出式（10-1）。

令 T_i 表示目的节点的秩达到 i 的时刻，$i=1, \cdots, K$，则分组投递延迟为 T_K。对于 T_i 的累积分布函数 $F_i(t):=P_r(T_i<t)$，以下 ODE 方程成立：

㊀ 这意味着，当两个节点 u 和 v 相遇时，只要 u 的秩大于 0 且 v 尚未满秩，u 就会一直向 v 发送随机线性组合（从节点 v 到节点 u 的传输同理）。

㊁ 即另一个节点的所有的编码数据包都是该节点的编码数据包的线性组合。

$$\frac{\mathrm{d}F_1(t)}{\mathrm{d}t} = \lambda\left(\sum_{j=1}^{B} X_j(t) + 1\right)(1 - F_1(t))$$

$$\frac{\mathrm{d}F_i(t)}{\mathrm{d}t} = \lambda\sum_{j=1}^{B} X_j(t)(F_{i-1}(t) - F_i(t)), i = 2, \cdots, K$$

其初始条件为 $F_i(0)=0$，$i=1$，…，K。

对于非编码方案，为了描述两个相遇的节点都有对对方有用的信息（即新生数据包）的概率，先假设秩为 i 的节点所携带的 i 个数据包（$i=1$，…，B）均是以相同的概率从 K 个原始数据包的集合中抽取得到。研究结果表明，携带 i 个数据包的节点能够从携带 j 个数据包的节点中获得一个新生数据包的概率 $P_r(i, j)$ 为

$$P_r(i, j) = \begin{cases} 1, & i<j, \\ 1-\binom{i}{j}\Big/\binom{K}{j}, & i\geqslant j \end{cases}$$

通过引入 $P_r(i, j)$，可以对式（10-1）～式（10-3）中的 ODE 修正为对非编码方案的建模方程：

$$\frac{\mathrm{d}X_i}{\mathrm{d}t} = \lambda\sum_{j=1}^{B} P_r(i-1,j)X_j(t)X_{i-1}(t)$$

$$-\lambda\sum_{j=1}^{B} P_r(i,j)X_j(t)X_i(t)i = 1,\cdots,B-1$$

参考文献［32，33］中对上述 ODE 方程进行了数值求解，得到了非编码方案和编码方案下的平均投递延迟，并将其与仿真结果进行了对比。在非编码方案下，文献中仿真了 3 种不同的调度策略：随机策略、本地最少策略以及全局最少策略。图 10-10 中比较了通过仿真得到的平均分组投递延迟和使用 ODE 模型预测的结果，图中所示的是在不同的网络节点总数量（不包括源节点）N 下的分组投递延迟。我们观察到，对于 RLC 方案和非编码方案，ODE 模型均低估了网络的分组投递延迟。考虑到前面所做的乐观假设，这个结果是在预期之中的。在上述 3 种调度策略中，全局最少策略假设节点具有关于网络中所有数据包的副本数量的全局信息，它是一种预言性方案。它的性能是最好的，随机策略次之，本地最少策略的性能最差。仿真结果还表明，不依赖于任何预言的 RLC 方案能够实现与基于预言的全局最少策略同样良好的性能。

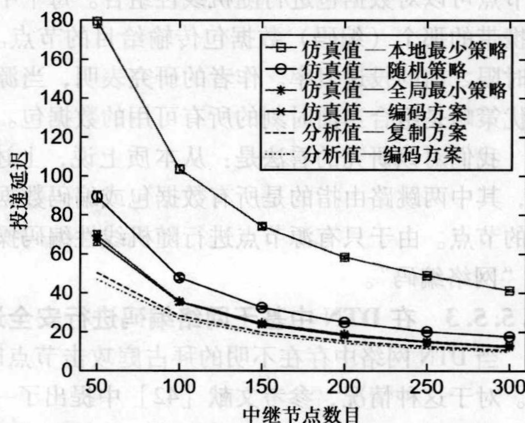


图 10-10　不同节点数量下的分组投递延迟（见参考文献［31］中的图 10-3）

10.5.5　其他关于在单播应用中使用 RLC 方案的研究工作

本节中，将简要地回顾一些其他关于将网络编码应用于 DTN 单播应用的研究工作。

10.5.5.1 优先编码协议

我们已经看到，在分组投递延迟（即交付整个分组所需的时间）方面，RLC 方案优于非编码方案。但是在平均数据包投递延迟（见图 10-5b）方面，RLC 方案不如非编码方案。对于不同数据包具有不同优先级的应用，参考文献［31］中提出了一种优先编码协议，其工作方式如下。假设源节点产生了 K 个目的节点相同的数据包，这 K 个数据包一共分为 M（$M<K$）个优先级，并满足：如果 $i<j$，则等级为 i 的数据包应先于等级为 j 的数据包被解码。首先，源节点使用 RLC 方案发送等级为 1 的数据包，对该等级的所有数据包都进行编码。只有在收到等级为 1 的数据包的 ACK 信号（由目的节点产生，使用 VACCINE 回收方案通过所有节点传播）后，源节点才会开始使用 RLC 方案发送等级为 2 的数据包。当网络中的节点接收到等级较高的数据包组合时，它们会丢弃本地存储的等级较低的数据包组合。在收到等级为 i 的数据包的 ACK 信号后，源节点会发送等级为 $i+1$（$i=1$，…，$M-1$）的数据包，直至目的节点解码出全部的 K 个数据包。

通过使用 10.5.4 节中的 ODE 模型，参考文献［32］中发现与非优先 RLC 方案相比，以上优先编码协议交付全部 K 个数据包所需的额外时间随优先等级总数的增大而线性增大。对于每个优先级，优先编码协议都会引入额外的延迟，该延迟等于投递该等级的第一个数据包所需的时间或是将 ACK 信息从目的节点传播到源节点所需的时间。

10.5.5.2 两跳方案中的最优控制

在参考文献［2］中作者考虑了以下情况：由一个源节点（在不同时刻）产生的 K 个数据包需要在一定时限之前被投递给一个目的节点。在将信息转发给中继节点之前，源节点可以对数据包进行随机线性组合。每个中继节点最多存储 1 个线性组合，且只把它携带的那个（编码）数据包传输给目的节点。研究目标是最大化这 K 个数据包在给定时限之前的投递概率。作者的研究表明，当源节点与某个中继节点相遇时，源节点的最优策略是组合当前时刻的所有可用的数据包。此外，作者还推导了投递概率的下界。

我们对该研究的看法是：从本质上说，上述路由方案是一个两跳路由下的 RLC 方案，其中两跳路由指的是所有数据包或编码数据包都可以经过一条 1 跳或 2 跳路径到达目的节点。由于只有源节点进行随机线性编码操作，因此从严格意义上说，该方案不算是“网络编码”。

10.5.5.3 在 DTN 中基于网络编码进行安全通信

当 DTN 网络中存在不明的拜占庭攻击节点时，这些节点会将损坏的数据包引入网络。对于这种情况，参考文献［42］中提出了一种基于 RLC 的安全单播路由协议。

考虑由一个源节点产生并去往一个目的节点的一组数量为 K 的数据包。假设攻击节点能够污染包括报头中的编码系数在内的整个编码数据包。目的节点接收到 K 个独立的编码数据包之后，由于这些编码数据包中的一部分可能会被污染，因此目的节点有可能解不出 K 个源数据包。

参考文献［42］中提出的协议结合了密钥分发和纠错编码，以保证数据包的安全传输。源节点将 K 个数据包编码为一个纠错编码向量空间[27]，并对编码数据包产生私

钥和公钥[58]，然后将公钥分发给网络中的某一部分节点。接收到公钥的节点都是安全的，它们可以使用公钥来验证接收到的随机组合。最后，源节点采用 RLC 方案发送编码数据包，并采用 E - NCP 重复控制机制。

在只考虑一个分代并且没有带宽和缓存限制的简单设定下，参考文献［42］中分析了上述协议，并从例如纠错编码向量空间的维度、需要接收公钥的节点的数量以及二进制喷射等待计数等协议参数方面描述了在给定时间内目的节点成功解码出 K 个数据包的概率。

10.6 开放性问题

本节将讨论一些值得进一步深入研究的开放性问题。

10.6.1 短消息应用的 RLC 效益

迄今为止的大部分工作研究的都是将线性网络编码应用于源节点成批产生的一组数量为 K 的数据包，并考虑交付整组数据包所需的时间（即分组投递延迟）。应用产生长信息时，多个数据包将分批到达。例如对于文件共享的情况。长信息需要被分割成多个更小的数据包，其部分原因是为了更多地利用短期传输机会。

对于应用产生短消息的情况，将短消息进一步分割为更小的数据包是低效的，这是因为发送更多的数据包意味着更多的低层协议开销。此外，RLC 方案的相对开销（存储并发送编码系数）为 $K\log_2(q)/P$，消息越短则 P 值越小，导致相对开销越大。

在这种情况下，是否使用以及何时使用随机线性编码或更一般的网络编码才能获得效益仍是一个有待解决的问题。假设数据包之间相互独立，那么需要考虑的不是分组投递延迟，而应是平均数据包投递延迟或顺序投递延迟。对于一个 RLC 方案来说，分代形成是其核心问题，即如何决定哪些数据包形成一个对之运用 RLC 方案的分代。最重要的是，我们已经看到，将来自不同单播流的数据包组合会降低 RLC 的效益（但对于缓存受限的情况它仍然有益）。对不同时刻产生的数据包（来自同一单播流）的不同组合选择仍有待研究。

10.6.2 多播通信中的 RLC 方案

多播通信支持将一个数据包分发给一组用户。相比于广播和单播的情况，网络编码在 DTN 多播通信中的应用受到的关注较少，部分原因是 DTN 多播通信中的不同的语义以及不同的多播算法[1, 6, 54, 59]。

对于带宽受限的基于 DTN 的内容分发网络，参考文献［23］中提出针对不同用户的不同兴趣进行自适应网络编码，只对类型相似的内容进行混合。初步的仿真结果表明，基于网络编码的方案能够实现低于非编码方案的解码延迟。此外，当只对类型相同的信息运用网络编码时，仅对小部分内容感兴趣的用户能够体验到更小的延迟，但代价是对广泛内容感兴趣的用户体验到的延迟较大。为了在网络编码方案带来的吞吐量提升

和延迟增大之间进行权衡，作者提出通过调整混合信息的方式使得类型不同但接收者有很多重叠的信息可以被混合。

由于本章中回顾的 RLC 方案能够在多播通信的两种极端情况（广播和单播）下表现出较好的性能，因此本章中的一些研究结果也可以应用于多播通信。要充分认识网络编码在 DTN 多播通信中的效益，还需要做进一步的研究。

10.7 总结与结束语

本章回顾了关于将网络编码应用于容断网络（DTN）的一些现有的研究成果。

首先介绍了在 DTN 广播和单播应用中基于非编码的传统路由方案，以及随机线性码（RLC）的基本操作，RLC 是文献中考虑较多的一种网络编码形式。接着介绍了 DTN 路由的设计空间，重点关注集成网络编码的不同选择。本章的重点是在单播通信和广播通信下对基于 RLC 的路由方案的性能评估，其中单播通信是大部分工作的研究对象。我们不仅重视理论结果，而且还强调了仿真研究结果。对于上述两种通信模型，基于 RLC 的方案能够改善能耗和延迟性能之间的权衡关系。

除了上述研究成果以外，我们还回顾了关于对 RLC 方案进行扩展的一些研究工作，例如优先路由以及安全通信的实现。最后，讨论了在我们看来需要进一步研究的开放性问题。

参考文献

[1] M. Abdulla and R. Simon. A Simulation Analysis of Multicasting in Delay Tolerant Networks. In Conference on Winter simulation (WSC), Monterey, CA, USA, 2006.

[2] E. Altman, F. De Pellegrini, and L. Sassatelli. Dynamic Control of Coding in Delay Tolerant Networks. In IEEE International Conference on Computer Communications (INFOCOM), Shanghai, China, 2010.

[3] A. Balasubramanian, B. N. Levine, and A. Venkataramani. DTN Routing as a Resource Allocation Problem. In ACM Conference on Communications Architectures, Protocols and Applications (SIGCOMM), 2007.

[4] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine. MaxProp: Routing for Vehicle - Based Disruption - Tolerant Networks. In IEEE International Conference on Computer Communications (INFOCOM), 2006.

[5] T. Camp, J. Boleng, and V. Davies. A Survey of Mobility Models for Ad Hoc Network Research. In Wireless Communications and Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Network Research: Research, Trends and Applications, 2002.

[6] M. C. Chuah and Y. Xi. An Encounter - based Multicast Scheme for Disruption Tolerant Networks. Technical Report LU - CSE - 07 - 009, Lehigh University, 2007.

[7] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. Introduction to Algorithms. The MIT Press, 1996.

[8] First Mile Solutions (i. e. , DakNet) . http: //firstmilesolutions. com/.

[9] S. Deb, M. M'edard, and C. Choute. Algebraic Gossip: A Network Coding Approach to Optimal Multiple Rumor Mongering. IEEE/ACM Transactions on Networking, special issue on networking and informationtheory, pages 2486 - 2507, 2006.

[10] A. Doria, M. Ud' en, and D. P. Pandey. Providing Connectivityto the Saami Normadic Community. In International Conference on Open Collaborative Design for Sustainable Innovation (dyd), Bangalore, India, December 2002.

[11] C. Fragouli, J. Widmer, and J. - Y. Le Boudec. Efficient broadcasting using network coding. IEEE/ACM Transactions on Networking, 16 (2): 450 - 463, April 2008.

[12] R. Groenevelt, P. Nain, and G. Koole. The message delay in mobile ad hoc networks. Performance Evaluation, 62 (1 - 4): 210 - 228, October 2005.

[13] Z. Haas, J. Halpern, and L. Li. Gossip - based Ad Hoc Routing. In IEEE International Conference on Computer Communications (INFOCOM), New York, NY, USA, 2002.

[14] Z. J. Haas and T. Small. A new networking model for biological applications of ad hoc sensor networks. IEEE/ACM Transactions on Networking, 14 (1): 27 - 40, February 2006.

[15] B. Haeupler. Analyzing Network Coding Gossip Made Easy (simpler proofs for stronger results even in adversarial dynamic networks) . In Allerton Conference on Communication, Control and Computing, Monticello, IL, USA, 2010.

[16] D. Hay and P. Giaccone. Optimal routing and scheduling for deterministic delay tolerant networks. In International Conference on Wireless On - Demand Network Systems and Services (WONS), Snowbird, Utah, USA, 2009.

[17] T. Ho, R. Koetter, M. M'edard, D. R. Karger, and M. Effros. The Benefits of Coding Over Routing in a Randomized Setting. In IEEE International Symposium on Information Theory (ISIT), Yokohama, Japan, June - July 2003.

[18] T. Ho, M. M'edard, R. Koetter, and D. R. Karger. A random linear network coding approach to multicast. IEEE Transactions on Information Theory, 52 (10): 4413 - 4430, 2006.

[19] P. Hui, A. Chaintreau, R. Gass, J. Scott, J. Crowcroft, and C. Diot. Pocket Switched Networking: Challenges, Feasibility, and Implementation Issues. In IFIP TC6 International Workshop on Autonomic Communication (WAC), 2005.

[20] B. Hull, V. Bychkovsky, Y. Zhang, K. Chen, M. Goraczko, A. K. Miu, E. Shih, H. Balakrishnan, and S. Madden. CarTel: A Distributed Mobile Sensor Computing System. In ACM Conference on Embedded Networked Sensor Systems (SenSys), New York, NY, USA, 2006.

[21] S. Jaggi, P. Sanders, P. A. Chou, M. Effros, S. Egner, K. Jain, and L. M. G. M. Tolhuizen. Polynomial time algorithms for multicast network code construction. IEEE Transactions on Information Theory, 51 (6): 1973 - 1982, 2005.

[22] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. - S. Peh, and D. Rubenstein. Energy - Efficient Computing for Wildlife Tracking: Design Tradeoffs and Early Experiences with ZebraNet. In ACM International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS), San Jose, CA, USA, 2002.

[23] G. Karbaschi and A. C. Viana. A Content – based Network Coding to Match Social Interest Similarities in Delay Tolerant Networks. In First Extreme Workshop on Communication (ExtremeCom), Laponia, Sweden, 2009.

[24] S. Katti, D. Katabi, W. Hu, and R. Hariharan. The Importance of Being Opportunistic: Practical Network Coding For Wireless Environments. In Allerton Conference on Communication, Control, and Computing, Montecello, IL, USA, 2005.

[25] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard, and J. Crowcroft. XORs in the air: Practical wireless network coding. IEEE/ACM Transactions on Networking, 16 (3): 497 – 510, 2008.

[26] D. Kempe, J. Kleinberg, and A. Kumar. Connectivity and inference problems for temporal networks. In Journal of Computer and System Sciences, Special issue on STOC 2000, 64 (4): 820 – 842, 2002.

[27] R. Koetter and F. Kschischang. Coding for errors and erasures in random network coding. IEEE Transactions on Information Theory, 54 (8): 3579 – 3591, August 2008.

[28] A. Krifa, C. Barakat, and T. Spyropoulos. Optimal buffer management policies for delay tolerant networks. In IEEE Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON), San Diego, CA, USA, 2007.

[29] R. Lidl and H. Niederreiter. Finite Fields, 2nd edition. Cambridge, England: Cambridge University Press, 1997.

[30] Y. Lin, B. Li, and B. Liang. Differentiated Data Persistence with Priority Random Linear Codes. In International Conference on Distributed Computing Systems (ICDCS), Toronto, Canada, 2007.

[31] Y. Lin, B. Li, and B. Liang. Efficient network coded data transmissions in disruption tolerant networks. In IEEE International Conference on Computer Communications (INFOCOM), Phoenix, AZ, USA, 2008.

[32] Y. Lin, B. Li, and B. Liang. Stochastic analysis of network coding in epidemic routing. IEEE Journal on Selected Areas in Communications, Special Issue on Delay and Disruption Tolerant Wireless Communication Systems, 26 (5): 794 – 808, June 2008.

[33] Y. Lin, B. Liang, and B. Li. Performance modeling of network coding in epidemic routing. In ACM/SIGMOBILE International Workshop on Mobile Opportunistic Networking (MobiOpp), Puerto Rico, 2007.

[34] J. Liu, D. Goeckel, and D. Towsley. Bounds on the gain of network coding and broadcasting in wireless networks. In IEEE Conference on Computer Communications (INFOCOM), Anchorage, AK, USA, 2007.

[35] D. S. Lun, M. M′edard, T. Ho, and R. Koetter. Network Coding with a Cost Criterion. In International Symposium on Information Theory and its Applications (ISITA), Parma, Italy, 2004.

[36] D. S. Lun, N. Ratnakar, M. M′ edard, R. Koetter, D. R. Karger, T. Ho, E. Ahmed, and F. Zhao. Minimum – Cost Multicast over Coded Packet Networks. IEEE Transactions on Information Theory, 52 (6): 2608 – 2623, 2006.

[37] C. Avin, M. Borokhovich, and Z. Lotker. Tight Bounds for Algebraic Gossip on Graphs. IEEE International Symposium on Information Theory, Austin, TX, 2010.

[38] A. Maffei, K. Fall, and D. Chayes. In Ocean Instrument Internet: Using Disruption Tolerant Networking to Join Heterogeneous Oceanographic Instrumentation into a Single Network, Honolulu, HI, USA, 2006.

[39] M. Mitzenmacher and E. Upfal. Probability and computing: Randomized algorithms and probabilistic analysis. Cambridge Press, 2005.

[40] S. -Y. Ni, Y. -C. Tseng, Y. -S. Chen, and J. -P. Sheu. The Broadcast Storm Problem in a Mobile Ad Hoc Network. In ACM Internation Conference on Mobile Computing and Networking (Mobicom), Seattle, WA, USA, 1999.

[41] J. Partan, J. Kurose, and B. N. Levine. A Survey of Practical Issues in Underwater Networks. In ACM International Workshop on Under - Water Networks (WUWNet), Los Angeles, CA, USA, 2006.

[42] L. Sassatelli and M. M'edard. Network coding for delay tolerant networks with byzantine adversaries. Preprint 2009, arXiv: 09075488, http: //cdsweb. cern. ch/ record/1195977/.

[43] T. Small and Z. J. Haas. The Shared Wireless Infostation Model - A New Ad Hoc Networking Paradigm. In ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC), 2003.

[44] T. Small and Z. J. Haas. Resource and performance trade - offs in delay - tolerant wireless networks. In SIGCOMM Workshop on Delay Tolerant Networking (WDTN), 2005.

[45] T. Spyropoulos, K. Psounis, and C. Raghavendra. Efficient routing in intermittently connected mobile networks: The multiple - copy case. In ACM/IEEE Transactions on Networking, 16: 77 - 90, 2008.

[46] T. Spyropoulos, K. Psounis, and C. Raghavendra. Efficient routing in intermittently connected mobile networks: The single - copy case. In ACM/IEEE Transactions on Networking, 16: 63 - 76, 2008.

[47] T. Spyropoulos, K. Psounis, and C. S. Raghavendra. Spray and wait: An efficient routing scheme for intermittently connected mobile networks. In SIGCOMM Workshop on Delay Tolerant Networking (WDTN), 2005.

[48] A. Vahdat and D. Becker. Epidemic Routing for Partially Connected Ad Hoc Networks. Technical Report CS-200006, Duke University, April 2000.

[49] J. Widmer and J. - Y. Le Boudec. Network Coding for Efficient Communication in Extreme Networks. SIGCOMM Workshop on Delay Tolerant Networking (WDTN), 2005.

[50] J. Widmer, C. Fragouli, and J. - Y. Le Boudec. Energy - efficient broadcasting in wireless ad hoc networks. In IEEE Workshop on Network Coding, Theory, and Applications (NETCOD), Riva del Garda, Italy, 2005.

[51] Wizzy Digital Courier. http: //www. wizzy. org. za/.

[52] Y. Wu, P. A. Chou, and S. -Y. Kung. Minimum - Energy Multicast in Mobile Ad hoc Networks using Network Coding. In IEEE Information Theory Workshop, 2004.

[53] S. -Y. Kung Y. Wu, P. A. Chou. Information Exchange in Wireless Networks with Network Coding and Physical - Layer Broadcast. Technical Report MSR - TR - 2004 - 78, Microsoft, August 2004.

[54] Q. Ye, L. Cheng, M. C. Chuah, and B. D. Davison. OS – multicast: On – demand Situation – aware Multicasting in Disruption Tolerant Networks. In IEEE Vehicular Technology Conference (VTC), Montreal, Canada, 2006.

[55] X. Zhang, G. Neglia, J. Kurose, and D. Towsley. On the Benefits of Random Linear Coding for Unicast Applications in Disruption Tolerant Networks. In NETCOD (IEEE Workshop on Network Coding, Theory, and Applications), 2006.

[56] X. Zhang, G. Neglia, J. Kurose, and D. Towsley. Performance modeling of epidemic routing. Elsevier Computer Networks Journal, 51/10: 2859 – 2891, 2007.

[57] X. Zhang, G. Neglia, J. Kurose, D. Towsley, and H. Wang. Random Linear Coding for Unicast Applications in Disruption Tolerant Networks. Technical Report 7277, INRIA, 2010. http://hal.archivesouvertes.fr/inria-00494473/en/.

[58] F. Zhao, T. Kalker, M. M'edard, and K. J. Han. Signatures for content distribution with network coding. In IEEE International Symposium on Information Theory (ISIT), Nice, France, 2007.

[59] W. Zhao, M. Ammar, and E. Zegura. Multicasting in Delay Tolerant Networks: Semantic Models and Routing Algorithms. In ACM Sigcomm workshop on Delay – Tolerant Networking (WDTN), Philadephia, PA, 2005.

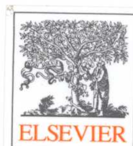




国际视野　科技前沿

国际信息工程先进技术译丛

《网络编码基础与应用》
《基于片上去耦电容的配电网络》（原书第2版）
《智能摄像机》
《短距离无线系统的可靠通信》
《车载系统和安全的数字信号处理》
《嵌入式系统设计 —— 嵌入式信息物理系统基础》（原书第2版）
《纳米封装 —— 纳米技术与电子封装》
《内容分发网络》
《全面的功能验证：完整的工业流程》
《无线Mesh网络架构与协议》
《UMTS蜂窝系统的QoS与QoE管理》
《半导体制造与过程控制基础》
《WCDWA原理与开发设计》
《下一代移动系统：3G/B3G》
《IMS:IP多媒体概念和服务》（原书第2版）
《下一代无线系统与网络》
《深入浅出UMTS无线网络建模、
《规划与自动优化：理论与实践》
《通信系统宽带无线接入》
《无线传感器及元器件：网络、设计与应用》
《印制电路板——设计、制造、装配与测试》
《IPTV与网络视频：拓展广播电视的应用范围》
《多电压CMOS电路设计》
《微电子技术原理、设计与应用》
《蜂窝网络高级规划与优化2G/2.5G/3G/…向4G的演进》
《基于蜂窝系统的IMS——融合电信领域的VoIP演进》
《无线网络中的合作原理与应用》
《电生理学方法与仪器入门》
《移动电视：DVB-H、DMB、3G系统和富媒体应用》
《环境网络：支持下一代无线业务的多域协同网络》
《基于射频工程的UMTS空中接口设计与网络运行》
《未来UMTS的体系结构与业务平台：全IP的3G CDMA网络》
《UMTS-HSDPA系统的TCP性能》
《宽带无线通信中的空时编码》
《数字图像处理》（原书第4版）





上架指导　通信技术／网络编码

ISBN 978-7-111-44506-7

定价：59.00元